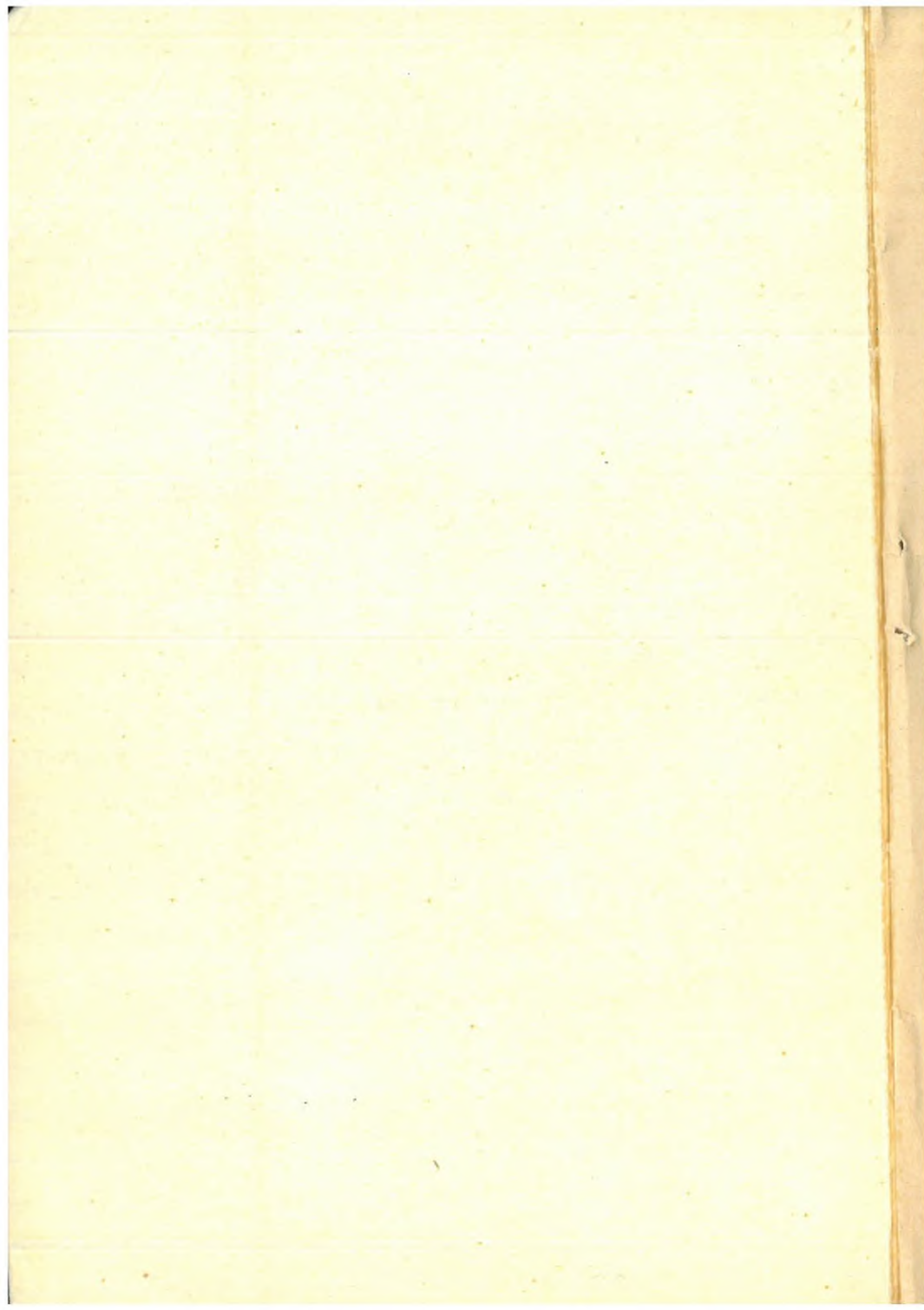


ESEM 
SOFTWARE

**PORADNIK OPERATORA
MINIKOMPUTERA
MERA-CAMAC-125/SM 4A**



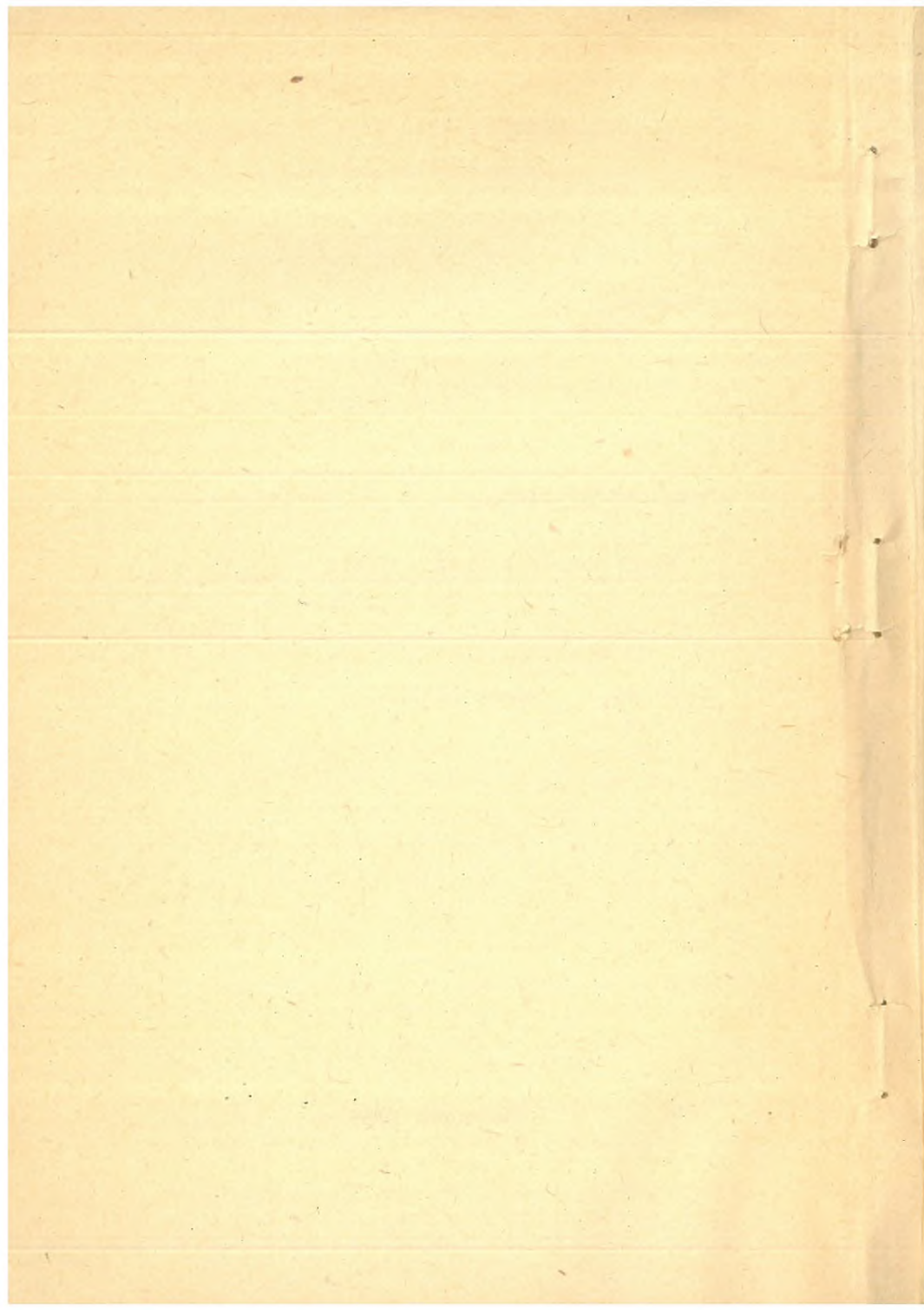
OŚRODEK DOSKONALENIA KADR TECHNICZNYCH
Rady Stołecznej NOT

PORADNIK OPERATORA

Minikomputera MERA - CAMAC - 125/SM 4 A

opracował: Marek LEWICKI

Warszawa 1985 r.



SPIS TREŚCI

	str.
BUDOWA MINIKOMPUTERA SM	11
1. Architektura systemów minikomputerowych SM	11
2. Procesor i jego funkcje	13
3. Przerwania, pułapki podprogramy	14
4. Organizacja i zarządzanie pamięcią	15
5. Pulpit procesora	16
5.1. Ładowanie systemu	17
6. Charakterystyka urządzeń zewnętrznych	18
6.1. Przykładowe urządzenia systemów MERA-SM	18
OPROGRAMOWANIE SYSTEMOWE MINIKOMPUTERÓW MERA-SM	21
1. Program obserwacji zasobów systemu - RMDEMO	28
1.1. Objaśnienie wyprowadzanych informacji	28
2. Opcje programu	30
PROGRAMY UŻYTKOWE SYSTEMU DOC PB	31
1. Wstęp	31
2. Format wiersza komendy systemu DOC PB	31
3. Format specyfikacji zbioru	31
4. Wywołanie programów użytkowych	33
5. Użycie zbiorów systemu DOC PB	33
PROCEDURY I KONWENCJE	35
1. Terminale	35
1.1. Znaki specjalne i znaki sterujące	35
1.2. Parametry terminala	41
1.2.1. Uprzywilejowanie terminala	41
1.2.2. Terminal przywiązany i nieprzywiązany	42
1.2.3. Terminal podporządkowany	42

	str.
1.3. Przynaglenia wejścia	43
1.3.1. Przynaglenie domniemane	43
1.3.2. Przynaglenie od zadania	43
1.3.3. Przynaglenie od procedury MCR	44
2. Urządzenia	46
2.1. Pseudourządzenia	48
2.2. Urządzenie zerowe	49
2.3. Urządzenie logiczne	50
3. Funkcje zabezpieczenia wieloużytkownikowego	51
3.1. Rozpoczynanie i kończenie pracy z systemem	51
3.1.1. Komenda HELLO	52
3.1.2. Komenda BYE	53
3.2. Urządzenia ogólnodostępne, urzędnia prywatne i niczyje	54
3.3. Program aktualizacji katalogu użytkowników systemu	56
3.3.1. Tworzenie katalogu użytkowników (opcja C).	57
3.3.2. Rozszerzanie katalogu (opcja A)	58
3.3.3. Usuwanie kont (opcja D).	60
3.3.4. Modyfikowanie kont przez nieuprzywilejowanych użytkowników	61
4. Program wyłączania systemu operacyjnego	62
KOMENDY MONITORA	67
PROGRAM PRZETWARZANIA ZBIORÓW KOMEND	80
1. Procesor zbiorów komend MCR	81
2. Symbole	84
2.1. Symbole specjalne	85
2.1.1. Logiczne symbole specjalne	85
2.1.2. Liczbowe symbole specjalne	86
2.1.3. Tekstowe symbole specjalne	87
2.2. Podstawienie wartości symbolu	87
2.3. Symbole i wyrażenia liczbowe	88
2.3.1. Symbole tekstowe, fragmenty tekstu i wyrażenia	89

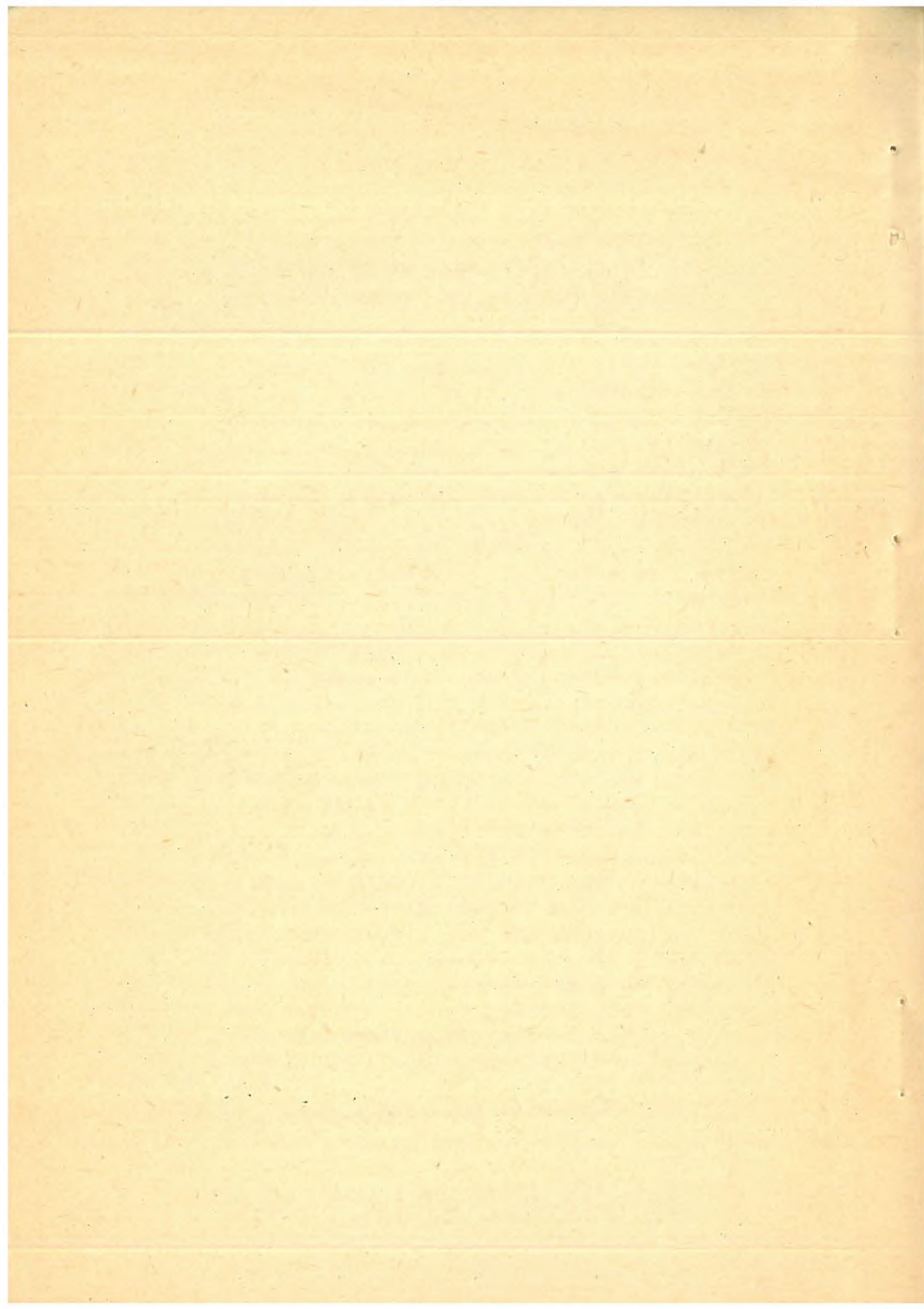
	str.
3. Klucze	90
4. Dyrektywy	90
5. Opis wybranych dyrektyw	91
KOMUNIKACJA Z UŻYTKOWNIKIEM 115	
1. Zadanie	115
2. Partycje	115
2.1. Typy partycji	115
2.2. Podpartycje	116
3. Wieloprogramowość	116
4. Tworzenie zadania	117
4.1. Instalowanie zadania	117
WPROWADZENIE 119	
1. Sposoby korzystania z edytora wierszowego	119
1.1. Metody wywoływania EDI	119
1.1.1. Specyfikacja zbioru wejściowego	120
1.2. Tryby sterujące: wprowadzania i wydawania komend	121
1.3. Tryby dostępu do tekstu	122
1.4. Konwencje komend edytora	123
1.4.1. Wykorzystanie gwiazdki (*)	123
1.4.2. Wykorzystanie wielokropka (...) w miejsce szukanego tekstu . .	123
1.4.3. Skróty komend	124
WPROWADZENIE 126	
1. Cele rejestracji błędów	126
2. Uruchamianie rejestratora błędów ERRLOG	128
3. Uruchamianie wstępnego formatera PSE .	129
4. Uruchomienie generatora sprawozdań SYE	130
4.1. Wiersz komendy zadania SYE . . .	131
5. Uruchamianie zadania wyłączenia rejestracji ERF.	135
6. Sprawozdania o błędach urządzeń . . .	136
7. Sprawozdanie zbiorcze	138

	str.
PROGRAM MANIPULACJI ZBIORAMI	143
1. Wiersz komendy PIP	143
1.1. Klucze i podklucze programu PIP	145
1.1.1. Klucze	145
1.1.2. Podklucze	145
1.2. Konwencja gwiazdki w specyfikacji zbioru	145
1.2.1. Symbol (#) w specyfikacji zbioru wyjściowego	146
1.2.2. Symbol (#) w specyfikacji zbioru wejściowego	147
2. Komendy programu PIP	147
2.1. Kopiowanie zbiorów systemu DOC PB	147
2.2. Funkcje sterujące programem PIP	152
PROGRAM FORMATOWANIA DYSKÓW	158
1. Rozpoczęcie i zakończenie pracy formatera	159
2. Tryby pracy formatera	160
2.1. Tryb automatyczny	160
2.2. Tryb manualny	161
3. Urządzenia formatowane przez FMT	162
3.1. DK: (dyski SMRK5/SMRK5F)	163
3.2. DP: (dyski SMRPR2/SMRP2/SMRP3)	163
4. Klucze formatera	164
4.1. /BAD	164
4.2. /DENS	164
4.3. /ERL	164
4.4. /MAN	165
4.5. /OVR	165
4.6. /VE	166
4.7. /WLT	166
4.8. /@Y	166
PODSTAWOWE KOMENDY EDYTORA EDI	167
1. Komendy sterujące	170
2. Komendy lokalizujące /sterujące wskaźnikiem wiersza/	171

	str.
3. Komendy modyfikacji i manipulacji tekstem	175
4. Makrokomendy	177
5. Komendy wprowadzania do lub ze zbioru	178
6. Komendy wydruku	180
7. Komendy zamykania zbiorów	181
8. Komunikaty informacyjne na poziomie komendy	183
PROGRAM TESTOWANIA URZĄDZEŃ ZEWNĘTRZNYCH IOX	
1. Wstęp	189
2. Zestawienie komend IOX	190
3. Uruchamianie IOX	193
4. Tryby działania IOX	194
5. Ustawianie parametrów czasu wykonania	196
6. Wydruki programu IOX	198
6.1. Raporty aktywności	198
6.2. Raporty błędów	199
7. Wyświetlanie informacji o stanie IOX	199
8. Sposób użycia IOX ze zbiorami komend	201
9. Opisy komend IOX	201
PROGRAM KONWERSJI ZBIORÓW FLX	
1. Wiersz komendy FLX	213
2. Klucze FLX	213
2.1. Klucze formatu tomu	214
2.2. Klucze formatu zbioru	214
2.3. Klucze sterujące	217
3. Działanie z taśmą papierową	220
PROGRAM TESTOWANIA NOŚNIKÓW DYSKOWYCH	
1. Wprowadzenie	222
2. Wywołanie programu BAD	222
2.1. Klucze programu BAD	223
2.2. Sprawdzenie nośnika	223
2.3. Współdziałanie komendy INI i programu BAD	224
3. Użycie programu BAD	224

	str.
PROGRAM WERYFIKACJI STRUKTURY	
ZBIORÓW VFY	226
1. Wprowadzenie do programu VDY	226
2. Wiersz komendy VFY	227
3. Klucze programu VFY	228
3.1. Kontrola poprawności struktury /funkcja domniemana/	228
3.1.1. Raportowanie błędów w zbiorach	229
3.1.2. Zbiory oznaczone do usunięcia	230
3.1.3. Usuwanie bloków wielokrotnie zaalokowanych	231
3.1.4. Eliminacja "wolnych" bloków .	231
3.1.5. Przywracanie bloków zgubionych	231
3.2. Klucz /DE (DELETE)	232
3.3. Klucz /FR (FREE)	232
3.4. Klucz /LI (LIST)	232
3.5. Klucz /LO (LOST)	233
3.6. Klucz /RC (READ CHECK)	233
3.7. Klucz /RE (REBUILD)	234
3.8. Klucz /UP (UPDATE)	234
PODSYSTEM WYPROWADZANIA PRT&QMG	236
1. Komenda PRINT i zarządzanie kolejką - wprowadzenie	236
2. Komenda PRINT	238
3. Nieuprzywilejowane komendy programu zarządzania kolejką	243
3.1. Komenda QUEUE /LIST	244
3.2. Komenda QUEUE /MOD	247
3.3. Komenda QUEUE /HOLD	248
3.4. Komenda QUEUE /RELEASE	249
3.5. Komenda QUEUE /DELETE	249
4. Instalowanie Podsystemu Wyrowadzania	250
4.1. Opisowe wprowadzenie i przegląd	251
4.2. Przykład komend uruchomienia Podsystem Wyrowadzania	254
5. Opisy uprzywilejowanych komend QMG	254
5.1. STOP	254
5.2. START	256

	str.
5.3. SPOOL	258
5.4. UNSPOOL	259
5.5. ASSIGN	260
5.6. DEASSIGN	260
6. Komunikaty Podsystemu Wyprowadzania .	261
PROGRAM KOPIOWANIA I KOMPRESJI DYSKÓW	262
1. Urządzenia obsługiwane przez DSC	262
2. Wywołanie i zakończenie DSC w wersji pod systemem	263
3. Wywołanie i zakończenie samodzielnej wersji DSC	263
4. Format komendy programu DSC	264
5. Deklaracja etykiet zbiorów i klucze programu DSC	265
5.1. Klucz sprawdzenia (/VE)	267
6. Klucze wersji samodzielnej programu DSC	268
PROGRAM ARCHIWACJI TOMÓW	270
1. Klucze komend i ich domniemane wartości	271
1.1. Komendy selektywnej archiwacji i odtwarzania	272
1.2. Komendy sterujące obsługą dysków	275
1.3. Klucze sterujące obsługą taśmy .	276
1.4. Klucze weryfikacji	277
1.5. Klucze listowania	277
1.6. Klucze inicjacji dysku	278
1.7. Urządzenia obsługiwane przez program BRU	279
2. Format wiersza komendy programu BRU .	280
2.1. Opis wiersza komendy	280
2.2. Opis przynagłeń	280
2.3. Opis parametrów wierszy komend .	281
2.3.1. Komenda bez specyfikacji zbiorów	283
2.3.2. Konwencja gwiazdka	283



BUDOWA MINIKOMPUTERA SM

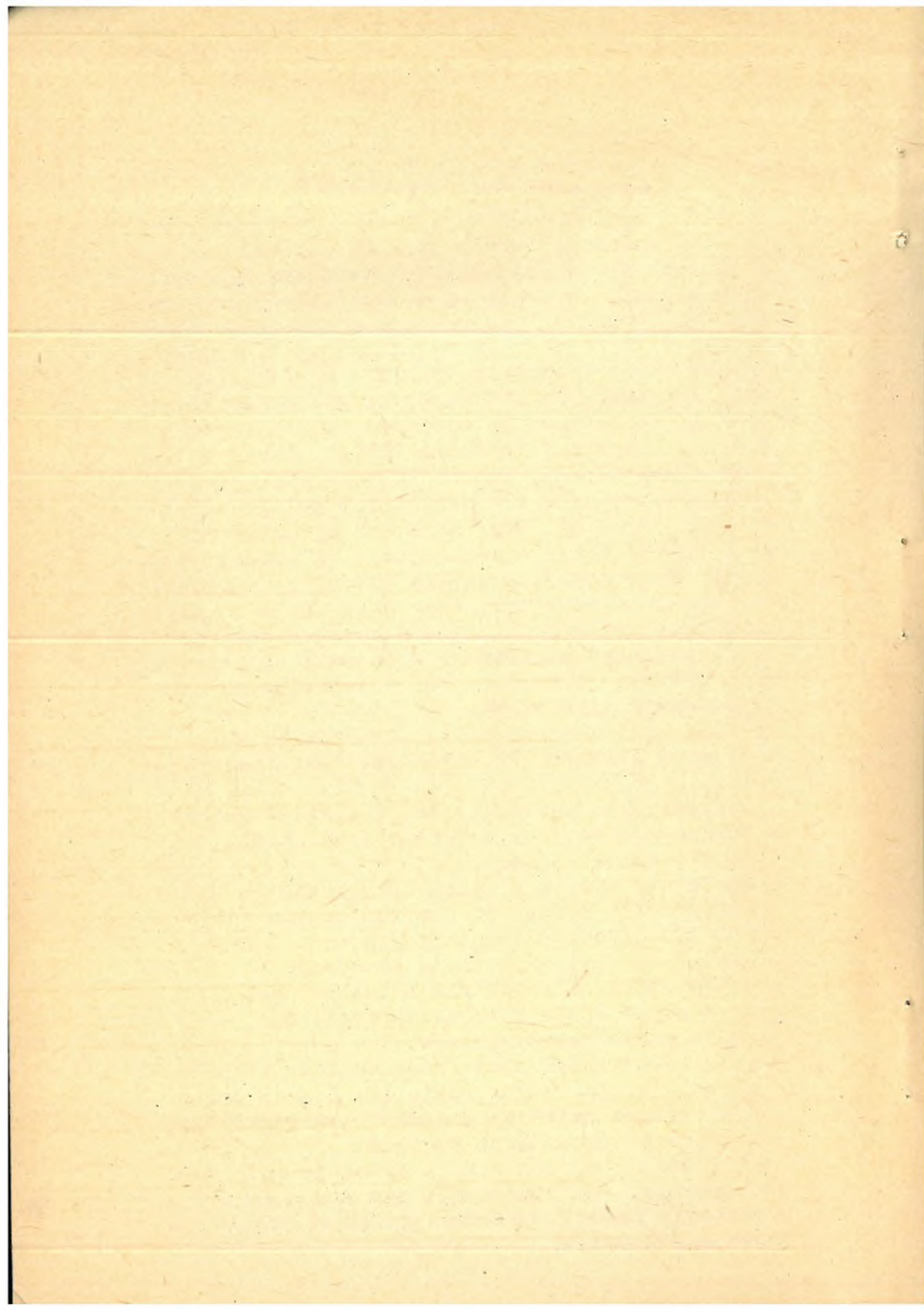
1. Architektura systemów minikomputerowych SM

SYSTEMY minikomputerowe typu SM są średniej wielkości systemami minikomputerowymi wyposażonymi w pamięć operacyjną o maksymalnej pojemności /zależnie od typu zastosowanego procesora/ od 28K do 124K słów szesnastobitowych. Składają się one z procesora, pamięci operacyjnej oraz zestawu urządzeń zewnętrznych dołączanych poprzez kontrolery do standardowej magistrali systemu nazywanej "WSPÓLNA SZYNA". Szyną tą są przesyłane adresy, dane i sygnały sterujące w identyczny sposób dla wszystkich modułów dołączonych do szyny. Wszystkie moduły dołączone do WSPÓLNEJ SZYNY - wliczając w to komórki pamięci operacyjnej, rejestry procesora i rejestry urządzeń zewnętrznych - mają swoje adresy. Wszystkie rozkazy działające na komórkach pamięci mogą być używane do działań na rejestrach urządzeń zewnętrznych, eliminując potrzebę rozkazów typu WE/WY.

Podstawowymi cechami minikomputerów SM są:

- słowo o długości 16 bitów /dwa 8-bitowe bajty/,
- bezpośrednia adresacja do 32 K słów $K=1024$,
- rozkazy jedno, dwu i bezadresowe działające na słowach i bajtach,
- sprzętowy stos w pamięci operacyjnej,
- bezpośredni dostęp do pamięci operacyjnej DMA dla urządzeń zewnętrznych,
- wektorowy system przerwań procesora,
- 8 wewnętrznych rejestrów uniwersalnych,
- arytmetyka stała i zmiennoprzecinkowa,
- zegar czasu rzeczywistego,
- sprzętowe rozszerzenie adresacji do 18 bitów pozwalające na adresowanie 124 K słów /oraz 4 K słów na rejestry urządzeń zewnętrznych/,
- modułarna konstrukcja sprzętu.

Dzięki modułowej konstrukcji istnieje możliwość elastycznego konfigurowania systemu w celu dostosowania zestawu urządzeń do określonych potrzeb użytkownika.



2. Procesor i jego funkcje

W systemach minikomputerowych SM, obecnie produkowanych w FMIK "ERA" stosowane są procesory produkcji radzieckiej klasy SM-4 tzn. CM 2104 i Elektronika 100-25 oraz klasy SM3 tzn. CM 1300. Moduł procesora dołączony do WSPÓLNEJ SZYNY wykonuje wszystkie operacje arytmetyczne i logiczne inicjowane przez rozkazy programu; za pomocą układu priorytetu szyny sprawuje nadzór nad przepływem informacji i przydziałem szyny; zapewnia przydział i ochronę pamięci operacyjnej aktualnie wykonywanym programom. Procesor SM4 może pracować w dwóch trybach:

- USER
- KERNEL

W trybie KERNEL procesor wykonuje wszystkie zlecenia programu. W trybie USER możliwości programu są ograniczone: część rozkazów jest niedozwolona, a dostęp do niektórych rejestrów zakazany. Konstrukcja taka pozwala na stosowanie wielodostępnych systemów operacyjnych zapewniających ochronę dostępu do zasobów systemu. Stosowane w minikomputerach SM procesory mają osiem rejestrów uniwersalnych, dostępnych dla programu, które mogą być wykorzystywane jako akumulatory, rejestry indeksowe, rejestry automodyfikacji lub wskaźniki stosu. Rejestr R7 jest wykorzystywany jako licznik rozkazów PC i zawiera adres następnego wykonywanego rozkazu.

Rejestr R6 jest wykorzystywany jako wskaźnik stosu SP wskazujący na ostatnie zajęte miejsce na stosie. Istnieją dwa niezależne wskaźniki stosu dla trybu KERNEL i USER.

Rejestry R0, R1, R2, R3, R4, R5 nie mają ściśle określonych funkcji i mogą być dowolnie wykorzystywane w programie.

Procesor posiada również rejestr zwany słowem stanu procesora (PSW) o adresie 777776/8/. Rejestr ten odzwierciedla aktualny stan procesora i zawiera następujące informacje:

1. Aktualny i poprzedni tryb pracy procesora,
2. Poziom priorytetu procesora względem przerwania zewnętrznych,
3. Bit T - ustawienie którego powoduje wystąpienie przerwania programowego po zakończeniu wykonania rozkazu. Mechanizm ten stosowany jest do śledzenia wykonania programów podczas poprawiania.
4. Kody warunków - opisują rezultat ostatnio wykonanego rozkazu. Bity NZVC ustawiają się przy wykonaniu rozkazów procesora.

3. Przerwania, pułapki podprogramy

Procesor obsługuje dwa rodzaje przerwania:

1. Zewnętrzne, przerwania od urządzeń zewnętrznych
 2. Wewnętrzne, zwane także pułapkami. Są to przerwania powodowane rozkazami procesora lub występują w wyniku nienormalnych stanów.
- Z punktu widzenia oprogramowania obsługa przerwania wewnętrznych i zewnętrznych jest podobna. Zgłoszone przerwania może być obsłużone po zakończeniu wykonywania bieżącego rozkazu. W przypadku zarejestrowania przez procesor zgłoszenia przerwania ustalony jest adres wektora przerwania. Dla pułapki adres ten jest bezpośrednio określony przez rodzaj pułapki. Dla przerwania od urządzenia zewnętrznego adres ten podawany jest przez kontroler urządzenia. Następnie na stos sprzętowy /określony przez rejestr SP/ w pamięci operacyjnej odkładany jest kolejno licznik rozkazów PC; słowo stanu procesora PSW. Stos rozbudowywany jest w kierunku malejących adresów pamięci. SP wskazuje zawsze na ostatnią zajętą pozycję na stosie. Następnie pierwsze słowo wektora przerwania ładowane jest do licznika rozkazów PC, a drugie słowo do PSW. Wektor przerwania wskazuje na program obsługi przerwania i określa warunki jego wykonywania.

Każde zgłoszenie przerwania wewnętrznego powinno być obsłużone. Kolejność obsługi pułapek i przerw wewnętrznych jest z góry ustalona. Kolejność obsługi przerw zewnętrznych zależy od priorytetu urządzenia, i aktualnego priorytetu procesora umieszczonego w PSW /liczba od 0 do 7/. Każdy kontroler urządzenia ma ustalony poziom przerwania /od 4 do 7/. Przerwanie zewnętrzne jest skuteczne tylko w tym przypadku jeżeli ma ono najwyższy priorytet z jedno - cześnie zgłaszanych, a ponadto jest on wyższy od aktualnego priorytetu procesora.

Stos sprzętowy wykorzystywany jest także w wywołaniu podprogramów. Wykonanie rozkazu skoku do podprogramu powoduje posłanie na stos jednego słowa opisującego kontekst programu /może być PC lub inny rejestr uniwersalny/, a załadunek do PC adresu podprogramu.

4. Organizacja i zarządzanie pamięcią

Pamięć operacyjna minikomputerów SM jest traktowana jako zbiór komórek o indywidualnych adresach, przy czym w każdej komórce jest pamiętane 16-bitowe słowo.

Słowa adresowane są zawsze parzyście. Młodszy bajt słowa ma ten sam adres co słowo. Starszy bajt słowa ma adres o jeden większy od adresu słowa. Program ma dostęp do słowa lub do bajtu w zależności od typu użytego rozkazu /bajtowy lub słowowy/. Ponieważ procesor posługuje się adresami 16 bitowymi możliwa jest bezpośrednia adresacja 32 K słów /tzn. 64 K bajtów/. Interfejs WSPÓLNA SZYNA daje możliwość adresacji do 128 K słów /18 bitów/. Dostęp do obszaru pamięci o adresach przekraczających adresowanie bezpośrednie jest uzyskiwany poprzez wykorzystywanie jednostki zarządzania pamięcią operacyjną MMU /memory management unit/ znajdującej się w procesorze. Zadaniem jednostki zarządzania pamięcią jest przekształcanie wirtualnej prze-

strzeni adresowej programu w fizyczną przestrzeń adresową pamięci operacyjnej i rejestrów urządzeń. Translacja adresów wirtualnych na fizyczne odbywa się w następujący sposób:

1. wirtualna przestrzeń adresowa dzielona jest na odcinki 4 K słów zwane stronami
2. z każdą stroną związany jest jeden 32-bitowy rejestr zwany rejestrem strony aktywnej APR. Składa się on z dwóch rejestrów 16-to bitowych: rejestru adresu strony PAR i rejestru opisu strony PDR. Są dwa zestawy rejestrów APR, po 8 dla każdego trybu pracy procesora (KERNEL i USER). Rejestr PAR podaje bazę przemieszczenia strony. Rejestr PDR podaje długość strony /może być krótsza niż 4 K/ oraz jej stan i sposób dostępu.
3. adres fizyczny tworzy się poprzez dodanie do adresu wirtualnego bazy przesunięcia jego strony.

5. Pulpit procesora

Procesor typu SM /zarówno SM-4 jak i Elektronika 100-25/ posiada pulpit wyposażony w szereg lampek i kluczy. W przypadku poprawnej pracy procesora służy on głównie do włączania zasilania systemu minikomputerowego, ładowania systemu i wyłączenia zasilania.

W wypadku uszkodzenia procesora lub któregoś z urządzeń zewnętrznych, może być pomocny w określeniu jego rodzaju i służy do ładowania systemu testowego lub wprowadzania małych programów diagnostycznych /bezpośrednio z kluczy/.

18 diod świecących pokazuje fizyczny adres WS. Umieszczone poniżej 16 diod wyświetla zawartość komórki pamięci o adresie podanym powyżej. Klucze /kolejno od lewej do prawej pozwalają na:

1. wpisanie do rejestru adresowego adresu ustawionego na kluczach

2. wyświetlenie w rejestrze danych zawartości adresu ustawionego w rejestrze adresowym i zwiększenie zawartości rejestru adresowego o 2.
3. wznowienie wykonania programu od bieżącego adresu
4. zatrzymanie procesora
5. rozpoczęcie wykonywania programu od adresu ostatnio załadowanego do rejestru adresowego, inicjacja procesora i urządzeń (RESET)
6. zapis danych ustawianych na rejestrze klu - czy pod adresem ustawionym w rejestrze adresowym i zwiększenie zawartości rejestru adresowego o 2.

Nad kluczami umieszczone jest 6 lampek sygnalizujących stan procesora /tzn. praca w trybie USER, włączona jednostka zarządzania pamięcią, zatrzymanie procesora, zajmowanie WSPÓLNEJ SZYNY przez procesor lub urządzenie zewnętrzne, wykonywanie mikroprogramu/.

5.1. Ładowanie systemu

Procedura ładowania systemu operacyjnego lub testowego zależy od typu procesora. Procesory CM 1300 i Elektronika 100-25 wyposażone są w emulator pulpitu, który umożliwia wykonywanie szeregu operacji z terminala operatorskiego TTO:. Adresem startu emulatora jest 773000/8/. Po uruchomieniu emulator wypisuje na TTO znak dolara jako przynaglenie. Aby załadować system należy wprowadzić mnemonik urządzenia, na którym znajduje się nośnik z systemem, numer jednostki urządzenia oraz znak powrotu karetki. Na przykład DK1. Dopuszczalne są następujące mnemoniki urządzeń: dla dysku kasetowego jest to DK; dla dysku elastycznego jest to DX; dla taśmy magnetycznej MT a dla taśmy papierowej PR. W przypadku procesora CM 2104 w celu załadowania systemu należy załadować odpowiedni dla danego urządzenia adres programu ładowania. Możliwe jest ładowanie systemu tylko z jednostek

o numerze zerowym. Adresy programów są następujące: dla dysku kasetowego - 773110, dla dysku elastycznego - 773220 a dla taśmy magnetycznej - 773210.

6. Charakterystyka urządzeń zewnętrznych

Jak wspomniano powyżej urządzenia zewnętrzne dołączane są poprzez kontrolery. Każdy kontroler reprezentowany jest przez swoje rejestry dostępne programowo tak jak komórki pamięci w obszarze ostatnich 4K słów adresów wspólnej szyny /tzn. od 124 K do 126 K/. Urządzenia mogą mieć różną liczbę rejestrów. Zależy to od ich funkcji.

6.1. Przykładowe urządzenia systemów MERA-SM

Konfiguracje systemów minikomputerowych mogą być zestawione z produkowanych w krajach socjalistycznych urządzeń zewnętrznych, takich jak: terminale, drukarki, pamięci na dyskach elastycznych, dyskowe pamięci kasetowe i pakietowe, pamięci taśmowe, stacje we/wy taśmy papierowej, plotery, adaptery interfejsów pomiarowych i jednostki transmisji synchronicznej. Pamięć dyskowa kasetowa **NERA-9450** jest urządzeniem przeznaczonym do wprowadzania/wyprowadzania informacji do/z minikomputera oraz jej zapisu/odczytu na/z nośnika informacji. Nośnikiem informacji jest dysk magnetyczny w kasecie typu MERA-847 lub odpowiednik.

Parametry pamięci MERA-9450:

Liczba dysków	2 /dysk stały oraz wymienny/	100 TPI
Liczba powierzchni na dysku	2	
Liczba ścieżek na powierzchni	204	
Liczba sektorów na ścieżce	12	
pojemność jednego dysku	2,5 M bajta	ISM
Średni czas dostępu do ścieżki	35 ms	

olka 200 TPI

Pamięć na dyskach elastycznych **SP55DE** jest urządzeniem przeznaczonym do wprowadzania/wyprowadzania informacji do/z minikomputera oraz jej zapisu/odczytu na/z nośnika informacji. Nośnikiem informacji jest dysk magnetyczny elastyczny.

Parametry pamięci SP55DE:

Liczba dysków	2 /wymienne/
Liczba powierzchni na dysku	1 lub 2
Liczba ścieżek na powierzchni	//
Liczba sektorów na ścieżce	26
pojemność jednego dysku	256 K bajtów

Pamięć taśmowa CM 5300.01 jest urządzeniem przeznaczonym do wprowadzania/wyprowadzania informacji do/z minikomputera oraz jej zapisu/odczytu na/z nośnika informacji. Nośnikiem informacji jest taśma magnetyczna szerokości 0,5 cala na szpuli.

Parametry pamięci CM 5300.01:

Gęstość i sposób zapisu	800 b.p.i ,NRZI
Szybkość transmisji danych	10 KBajt/s
Liczba ścieżek	9
Prędkość robocza	0,3175 m/s
Maksymalna średnica szpuli	216 mm

Drukarka D-180 jest urządzeniem przeznaczonym do wyprowadzania informacji z komputera. Pozwala ona na drukowanie pełnego zestawu znaku ASCII z maksymalną szybkością 180 zn/s.

Parametry drukarki D-180:

Szybkość drukowania	50 lub 40 wierszy na minutę
Liczba znaków w wierszu	132 lub 158
Matryca	9 x 7 kropek
Maksymalna liczba odbitek	oryginał + 4 kopie
Szerokość papieru	od 4 do 17 cali

Terminal **MERA-7953N** jest urządzeniem przeznaczonym do wprowadzania/wyprowadzania informacji

do/z minikomputera. Wprowadzanie informacji następuje przy użyciu klawiatury a informacja wprowadzana wyświetlana jest na ekranie terminala.

Parametry terminala MERA-7953N:

Format	24 wiersze na 80 znaków
Szybkość transmisji	300,600,1200,4800,9600 bodow

Terminal jest funkcjonalnym odpowiednikiem VT-52 firmy DEC.

OPROGRAMOWANIE SYSTEMOWE MINIKOMPUTERÓW MERA-SM

Walory minikomputera MERA-SM ujawniają się w pełni podczas pracy pod nadzorem Dyskowego Systemu Operacyjnego Czasu Rzeczywistego DOC PB. System ten umożliwia sterowanie procesami, zbieranie danych pomiarowych, obliczenia inżynierskie i ekonomiczne, opracowywanie nowych programów itp. zastosowania.

System operacyjny DOC PB jest systemem generowanym dla konkretnej konfiguracji sprzętu i potrzeb eksploatacyjnych.

System DOS RW można wygenerować w dwu podstawowych odmianach:

- system mapowany - współpracuje z jednostką zarządzania pamięcią. Umożliwia wykorzystanie 124 k słów pamięci operacyjnej. Pamięć operacyjna podzielona jest na rejony podlegające ochronie dostępu. Umożliwia dynamiczną gospodarkę pamięcią. System mapowany nie może być użytkowany na procesorze SM-3 i jego odpowiednikach.
- system niemapowany - nie współpracuje z jednostką zarządzania pamięcią. Umożliwia wykorzystanie 28 k słów pamięci operacyjnej. Nie zapewnia ochrony dostępu i dynamicznej gospodarki pamięcią operacyjną.

Podstawowym pojęciem w systemie operacyjnym DOC PB jest ZADANIE. ZADANIE jest to program w postaci pozwalającej na jego bezpośrednie wykonanie w minikomputerze MERA-SM pod nadzorem systemu operacyjnego DOC PB.

Proces powstawania ZADANIA składa się z kilku etapów:

- użytkownik pisze program w języku symbolicznym (MAKROASSEMBLER, FORTRAN IV, COBOL, BASIC PLUS-2, PASCAL)
- użytkownik zapisuje tekst programu w postaci jednego lub więcej modułów w zbiorze źródłowym

- moduły programu są translowane na postać pośrednią i składowane w zbiorach modułów pośrednich. Postać pośrednia jest przesuwalna i nie ma ustalonych adresów pamięci operacyjnej
- z modułów pośrednich programu użytkownika i bibliotecznych modułów pośrednich budowane jest ZADANIE i składowane w zbiorze obrazu zadania.

ZADANIE ma ustalone adresy pamięci i nie jest przesuwalne.

ZADANIE może być segmentowane i posiadać nakładki ładowane automatycznie lub na żądanie.

- ZADANIE jest instalowane i wykonywane.

Z punktu widzenia ZADANIA system operacyjny DOC PB stwarza środowisko umożliwiające wykonanie ZADANIA. Wszystkie główne użytkowe funkcje systemu realizowane są przez wyspecjalizowane zadania systemowe.

Jądro systemu stanowi EGZEKUTOR, który nadzoruje wykonywanie wszystkich zadań w systemie DOC PB.

Zasadnicze cechy systemu DOC PB

System operacyjny DOC PB można scharakteryzować następującymi określeniami:

- wielozadaniowość; wiele zadań jednocześnie może być aktywnych; egzekutor systemu przydziela zadaniom zasoby sprzętu i czas procesora na podstawie wysokości priorytetu przyznanego zadaniom; priorytet wyrażony jest liczbą od 1 do 250 dziesiętnie,
- wielodostępność; wielu użytkowników jednocześnie może komunikować się interakcyjnie z systemem; dostęp do systemu i jego zasobów podlega ochronie programowej,

- działanie w czasie rzeczywistym;

krótki czas reakcji na zdarzenia zewnętrzne, możliwość przyznania wysokiego priorytetu zadaniom w istotny sposób zależnym od czasu, obsługiwany zestaw urządzeń /w tym CAMAC-SM/, dostępne dyrektywy systemowe i procedury, które mogą być wykorzystane w programach pisanych zarówno w języku MAKROASSEMBLER jak i FORTRAN w pełni potwierdzają to określenie.

Wartych podkreślenia jest kilka szczególnych cech systemu:

- rozbudowany podsystem wejścia-wyjścia
Umożliwia on w standardowej postaci transmisje danych na kilku poziomach organizacji:
 - a/ transmisje bloków logicznych o zmiennej długości,
 - b/ transmisje bloków wirtualnych i stałej długości,
 - c/ transmisje rekordowe /dostęp swobodny dla dysków i sekwencyjny/.

W przypadku a/ i b/ operacje wejścia-wyjścia wykonywane są asynchronicznie względem zadania i nie wytrzymują jego działania. Opcjonalnie dostępna jest w systemie mapowanym możliwość rozszerzonego zarządzania rekordami w zbiorach na dyskach i taśmach magnetycznych /RCS - record control service/. W rozszerzeniu względem standardowo stosowanych w DOC PB sposobów dostępu do rekordów, opcja ta zapewnia dostęp kluczowy /indeksowy/ i względny /relatywny/.

Dostęp do zbiorów jest w systemie DOC PB realizowany centralnie, automatycznie i w pełni bezpiecznie /bez możliwości zakłóceń pomiędzy użytkownikami/.

- przeładowywanie zadań
W systemach z ograniczoną pamięcią operacyjną i wieloma zadaniami EGZEKUTOR uwalnia pamięć operacyjną dla zadania o wysokim priorytecie wyładowując na dysk zadanie o priorytecie niskim.
Gdy zadanie o wysokim priorytecie przestanie być aktywne, to zadanie wyładowane będzie załadowane ponownie do pamięci i kontynuowane od miejsca zatrzymania.
- synchronizacja poprzez istotne zdarzenie
Istotnym zdarzeniem w systemie nazywa się zakończenie operacji wejścia-wyjścia, zakończenie zadania, usunięcie zadania z kolejki, wykonanie jednej lub szeregu dyrektyw systemowych, upłynięcie wyspecyfikowanego odcinka czasu.
Po wystąpieniu istotnego zdarzenia EGZEKUTOR ocenia i zmienia stan wszystkich zadań w systemie, przydziela zasobów, przydziela czasu procesora oraz porządkuje kolejki.
- bazowanie na dysku
Poszczególne elementy systemu operacyjnego ładowane są do pamięci operacyjnej tylko w czasie potrzeby.
W pozostałym czasie rezydują one w pamięci dyskowej.
Jedynym wyjątkiem jest EGZEKUTOR permanentnie rezydujący w pamięci operacyjnej.

Podstawowe funkcje systemu realizowane przez wyspecjalizowane zadania

SYSTEM ZBIORÓW

System operacyjny DOC PB posiada integralnie wbudowany system zbiorów tworzący strukturę zbiorów na dyskach twardych, elastycznych i taśmach magnetycznych. Zarządzanie zbiorami jest w pełni zautomatyzowane i ich fizyczna struktura jest poza kontrolą zwykłego użytkownika. Zapewniona jest także ochrona dostępu do zbiorów przed niepożądaną ingerencją.

KOMUNIKACJA Z UŻYTKOWNIKIEM

Użytkownik komunikuje się z systemem za pośrednictwem specjalnego zadania, które realizuje komendy operatorskie, wywołuje inne zadania systemowe oraz zapewnia ochronę systemu przed nieuprawnionym dostępem.

TRYB PAKIETOWEGO PRZETWARZANIA

Możliwe jest stworzenie zbiorów komendy, które mogą być wielokrotnie wywoływane i wykonywane, co pozwala na usprawnienie czynności rutynowych. Poza komendami operatorskimi zbiory te mogą zawierać dyrektywy zezwalające na wprowadzenie parametrów, zapętlenie, skoki, komunikację z operatorem itp.

SIECI KOMPUTEROWE

System DOC PB umożliwia łączenie wielu systemów minikomputerowych w sieci realizujące funkcje w sposób przestrzennie rozłożony /zdalne przetwarzanie zbiorów, zdalne użytkowanie urządzeń wejścia-wyjścia, komunikacja pomiędzy zadaniami w różnych systemach - węzłach sieci, zdalne wykonywanie zadań/.

Możliwe jest ponadto przyłączanie minikomputerów SM do systemów komputerowych JS-RIAD poprzez emulację terminali systemu RIAD na minikomputerach SM.

REJESTRACJA BŁĘDÓW SPRZĘTU

W czasie pracy systemu mogą wystąpić błędy i niesprawności sprzętu, które często są niewidoczne dla użytkownika.

Specjalne zadania systemowe rejestruje te błędy te błędy w zbiorze dyskowym, co daje możliwość późniejszego zaznajomienia się z nimi w celu oceny stanu sprzętu.

Systemowe zadania użytkowe

- translatory

System DOC PB zaopatrzony jest w rozbudowany MAKROASSEMBLER. Pozwala on na translację prog-

ramów zapisanych w postaci symbolicznych rozkazów procesora oraz na stosowanie dyrektyw systemowych. Ponadto pożliwe jest definiowanie i użytkowanie makrodyrektyw, posługiwa - nie się bibliotekami makrodyrektyw oraz translacja warunkowa.

Kompilator FORTRAN IV Systemu DOC PB posiada szereg istotnych rozszerzeń względem wzorca, szczególnie w zakresie operacji wejścia-wyjścia. Ponadto możliwym jest używanie wielu procedur zgodnych ze standardem ISA dla sterowania procesami i innych zastosowań w trybie czasu rzeczywistego.

Dostępne są także procedury ułatwiające współpracę z interfejsem CAMAC-SM. Interpreter BASIC jest także znacznie rozszerzony względem wzorca, szczególnie w zakresie operacji wejścia-wyjścia i możliwości dołączania procedur zewnętrznych. Możliwa jest także opcja kompilacyjna języka BASIC.

W mapowanym systemie DOC PB może być także użytkowany kompilator COBOL zgodny z amerykańską normą ANSI z 1974 r. Stosowanie kompilatora COBOL jest możliwe po wgenerowaniu w system DOC PB opcji rozszerzonego zarządzania rekordami RCS.

- edycja tekstów źródłowych

W systemie DOC PB używane są dwa edytory tekstów źródłowych:

EDI - przeznaczony do pracy interakcyjnej
/wierszowy/

KED - przeznaczony do pracy interakcyjnej
/ekranowy/.

Pozwalają one na zakładanie zbiorów źródłowych, ich aktualizację i poprawianie.

- budowa zadań

Do budowy zadań z modułów pośrednich i bibliotecznych przeznaczony jest "budowniczy zadań" TKB dostępny w trzech wersjach różniących się prędkością działania i zajętością pamięci.

INNE ZADANIA UŻYTKOWE

- LBR: program zarządzania bibliotekami modułów pośrednich i makrodyrektyw.
- PIP: program manipulowania zbiorami; służy do kopiowania, łączenia, usuwania, zmiany nazwy itp. operacji na zbiorach.
- ODT: program oczyszczania zadań; ułatwia uruchamianie zadań poprzez możliwość listowania zawartości komórek pamięci i ich modyfikację, śledzenie zmian zawartości komórek, zatrzymywanie i wznawianie wykonywania programu.
- DMP: program listowania zawartości dowolnych zbiorów w postaci symbolicznej, oktalnej słownej, oktalnej bajtowej i Radix-50.

W systemie istnieje ponadto kilka zadań systemowych do kopiowania nośników magnetycznych, sprawdzania fizycznej, logicznej i strukturalnej poprawności nośników magnetycznych oraz modyfikacji zbiorów modułów pośrednich i zbiorów obrazów zadań.

Minimalna konfiguracja sprzętu

Minimalna konfiguracja sprzętu, na której może funkcjonować system DOC PB składa się z następujących modułów:

- procesor SM-3, SM-4 lub ich odpowiedniki
- pamięć operacyjna 16 k słów
- terminal operatora
- pamięć dyskowa

Konfiguracja taka pozwala na proste zastosowanie systemu niemapowanego z jednym użytkownikiem. Powiększenie pamięci operacyjnej do 24 k słów pozwala na użytkowanie systemu mapowanego (za wyjątkiem procesora SM-3 i jego odpowiedników). Dalsze powiększanie pamięci operacyjnej umożliwi pracę wielozadaniową, a po dołączeniu kolejnych terminali pracę wielodostępną.

1. Program obserwacji zasobów systemu - RMDEMO

RMDEMO jest zadaniem uprzywilejowanym, które pokazuje informacje dotyczące aktywności zadania w systemie operacyjnym DOC PB-2. Informacje są zazwyczaj pokazywane na terminalu ze sterowaniem kursora, takim jak MERA 7953 VGD lub jego odpowiedniku. Jednorazowy obraz pamięci wyprowadzany jest na terminalach nie posiadających sterowania kursorem. Sposób wyprowadzania określony jest przez program na podstawie systemowej charakterystyki urządzenia. Umiejscowienie i rozmiary zadań, pokazywane przez RMDEMO, należy traktować jako przybliżone.

1.1. Objaśnienie wyprowadzanych informacji

DOC PB-2 BL26 124K 10-SEP-83 12:21:30
 IDLE POOL=1380.:2600.:45. FREE=SYO:246. DB1:
 PARS

	1380.:2600.45.	DB2:	DB3:	
IN:	" T M F	"	"	"
				LDPAR :T
5	" T C 1	"	"	"
				DRVPAR:D
45K	" : R 1	"	"	"
				SYSPAR:T
OUT:	L " A	K	R	T
				FCPPAR:D
0	D " C	5	M	K
				GEN :D
OK	R " P	2	D	B
	>--->--->----->----->-----			
	0###7###15###23###31###38###46###54###			
	E-----P---TD---T-D-----			

	62###69###77###85###93###100###108###111###			
	----->			
				>
				D
				X
				:
				ERRSEQ
				14.

Rys.1. Przykładowy rysunek wyprowadzany przez RMDEMO

1. Numer wersji systemu, poziom bazowy i rozmiar pamięci
2. Bieżąca data i czas
3. Aktualne wykonywane zadanie. «IDLE» oznacza, że żadne zadanie nie jest wykonywane.
4. Informacje o pamięci dynamicznej systemu.
POOL=X :Y :Z gdzie:
 X1:Y1:Z1
X - rozmiar największego wolnego bloku w puli (w słowach)
Y - całkowita liczba wolnych słów w puli
Z - liczba wolnych fragmentów w puli
X1,Y1,Z1 - wskazują najmniejsze wartości wykryte od czasu uruchomienia RMDEMO.
5. Liczba wolnych bloków na SYO: i trzech wybranych zamontowanych urządzeniach o strukturze zbiorów DOC-PB.
6. Liczba aktywnych zadań w pamięci, wraz z ilością pamięci przez nie zajmowaną. Liczba zadań aktywnych nie znajdujących się w danej chwili w pamięci (wyladowanych na dysk), wraz z łączną ilością pamięci jakiej one potrzebują.
7. Nazwy zadań znajdujących się w pamięci wyświetlane są pionowo, zaś pamięć zajmowana przez to zadanie przedstawiana jest nawiasami kwadratowymi i liniami pod (względnie nad) nazwą zadania. Postać nawiasów kwadratowych i linii użytych dla przedstawienia zajętości pamięci ma specjalne znaczenie.
{ } oznacza, że zadanie jest aktywne
[] oznacza, że zadanie nie jest aktywne, lecz zajmuje pamięć
! ! oznacza, wspólny obszar; pokazywana nazwa jest jego nazwą
+ + oznacza obszar wspólny bez nazwy, wyświetlana jest nazwa pierwszego przyłączonego zadania
() oznacza, załadowany ładowalny drajwer, wyświetlany jest mnemonik urządzenia

----- znaczy, że zadanie nie jest umiesz-
czone na stałe w pamięci
===== znaczy, że zadanie jest umieszczone
na stałe w pamięci

8. Ładowne drajwery, które są załadowane.
Zamiast nazwy zadania pokazywany jest mne-
monik urządzenia.
9. Partycje w systemie. Najpierw podana jest
nazwa partycji, a potem litera, reprezentu-
jąca typ partycji.
Litery są zdefiniowane jak następuje:
T = partycja zarządzana przez użytkownika
S = podpartycja
D = partycja zarządzana przez system
C = partycja wspólna
10. Rozmiar partycji i jej miejsce w pamięci.
Początek każdej partycji zaznaczony jest li-
terą. Oprócz liter opisanych w punkcie 9 do-
datkowo są używane:
E = egzekutor
P = obszar pamięci dynamicznej systemu
Gwiazdki umieszczone pomiędzy literami od-
zwierciedlają przybliżoną wielkość pamięci
zajmowaną przez partycje.
11. Liczba błędów wykrytych przez ERLOG.

2. Opcje programu

W programie RMDemo możliwa jest zmiana dwóch
parametrów wyświetlania:

- nazwy urządzenia, którego wolny obszar będzie
wyświetlany
- częstotliwości uaktualniania informacji wy-
świetlanej

Sposób wprowadzania komend modyfikujących pa-
rametry jest wyświetlany po naciśnięciu znaku
ESC w czasie wyświetlania przez RMDemo obrazu
pamięci.

PROGRAMY UŻYTKOWE SYSTEMU DOC PB

1. Wstęp

System operacyjny DOC PB posiada szereg programów użytkowych umożliwiających współpracę z różnymi typami zbiorów umieszczonych na różnych nośnikach /np. dyskach, taśmach magnetycznych i taśmach kasetowych/. Poniżej opisany jest format wiersza komendy, specyfikacja zbioru i sposób wywołania programów użytkowych.

2. Format wiersza komendy systemu DOC PB

Ogólny format wiersza komendy programów użytkowych w systemie DOC PB ma następującą postać:

outfil,...outfil=infil,...infil <CR>

gdzie outfil i infil są specyfikacjami zbiorów wyjściowych i wejściowych, które są przetwarzane przez program użytkowy.

3. Format specyfikacji zbioru

Format specyfikacji zbioru ma następującą postać:

dev:[grup,memb]filnam,filtyp;ver/sw.../subsw...
<CR>

dev: Urządzenie fizyczne zawierające wymagany tom. Nazwa zawiera 2 znaki ASCII, po których może wystąpić 3 cyfrowa liczba oktalna i dwukropek np. DKO:, TT101:.
Domniemanym jest pseudourządzenie SYO:.

[grp,memb] Numer grupy i numer użytkownika w grupie skojarzony z katalogiem(UFD) użytkownika zawierającym wymagany zbiór. Obydwie liczby są oktalne. Domniemanym jest bieżący UIC.

filnam Nazwa zbioru. Może zawierać od 0 do 9 znaków.
Nie ma domniemania.

- filtyp** Typ zbioru. Oddzielony jest od nazwy zbioru kropką. Może zawierać od 0 do 3 znaków.
- ver** Nie ma domniemania.
Liczba oktalna określająca różne wersje tego samego zbioru. Nowoutworzony zbiór ma wersję o numerze 1. Numer wersji może leżeć w zakresie od 0 do 77777/8/. Jest on oddzielony od typu zbioru średnikiem.
- /sw** Domniemana jest ostatnia wersja /o największym numerze/.
Nazwa określająca klucz związany z funkcją wykonywaną przez program użytkowy. Może być w następujących formatach:
/sw wywołuje funkcje klucza
/-sw neguje funkcje klucza
/NOSW neguje funkcje klucza
Klucze mogą przyjmować wartości liczbowe i tekstowe. Większość wartości liczbowych jest traktowana jako domniemana wartość oktalna. Aby podać wartość dziesiętną należy zakończyć liczbę kropką. Wartości poprzedzoną znakiem (#) są traktowane jako oktalne. Dowlolna wartość może być poprzedzona znakiem (+) lub (-).
Przykładowe specyfikacje kluczy:
/SW:27.:MAP:29.
/-SW
/NOSW:-#50:SWITCH
- /SUBSW** Nazwa określająca podklucz związany z danym kluczem. Identyfikuje ona podfunkcję odnoszącą się do funkcji słownego klucza. Na przykład:
PIP>[200,200]*.#/PR/FO<CR>
W tym przypadku, /FO jest podkluczem dla podklucza PR.
Reguły odnoszące się do kluczy są prawdziwe także dla podkluczy.

4. Wywołanie programów użytkowych

Programy użytkowe zainstalowane w systemie za pomocą komendy INSTALL programu MCR mogą być uruchamiane w następujący sposób:

1. Wywołanie programu i przekazanie sterowania do MCR:

>nazwa-programu wiersz-komendy <CR>
lub

MCR>nazwa-programu wiersz-komendy <CR>

Metoda ta pozwala na wykonanie pojedynczej komendy. Nie mogą być w ten sposób uruchomione programy SLP i ZAP.

2. Wywołanie programu i przekazanie sterowania do tego programu:

>nazwa-programu <CR>

Po wywołaniu program odpowiada przynagleniem: nazwa-programu >

można wtedy wprowadzić wiersz komendy, którą chcemy wykonać. Aby zakończyć wprowadzanie należy wprowadzić CTRL/Z.

Programy użytkowe niezainstalowane w systemie mogą być uruchomione w następujący sposób:

1. Komenda RUN:

>RUN Xnazwa-programu <CR>

Znak (X) określa katalog systemowy, który będzie przeglądany w celu znalezienia zadania.

2. Komenda RUN zmieniająca UIC wykonywania programu:

>RUN Xnazwa-programu /UIC=[grp,memb] <CR>

Po wywołaniu program odpowiada przynagleniem i czeka na wprowadzenie wiersza komendy. Wprowadzenie CTRL/Z kończy wykonywanie programu. *crty/c*

5. Użycie zbiorów komend

Zbiory komend zawierają sekwencje wierszy komend, które mogą być interpretowane przez pojedyncze zadanie. W celu wywołania takiego zbioru dla programu PIP należy wprowadzić:

>PIP@PIPCMDS.CMD<CR>

W tym przykładzie PIP wykonuje sekwencję komend zawartą w zbiorze PIPCMS.CMD.

Zbiory komend mogą zawierać komendy MCR. Tego typu zbiory są wywoływane w następujący sposób:

>@zbiór-komend<CR>

Domniemania specyfikacji zbioru komend:

- Urządzenie -- SYO:
- [grup,memb] -- bieżący UIC
- Nazwa zbioru -- musi być podana
- Typ zbioru -- .CMD
- Wersja -- ostatnia wersja zbioru.

PROCEDURY I KONWENCJE

Rozdział ten opisuje procedury i konwencje, które operator winien znać po to, by zarządzać systemem z terminala.

Opisane są następujące zagadnienia:

- jak obsługiwać terminal
- jak zarządzać urządzeniami peryferyjnymi

- jak pracować w systemie z zabezpieczeniem wieloużytkownikowym
- jak wyłączać system operacyjny DOC PB

1. Terminale

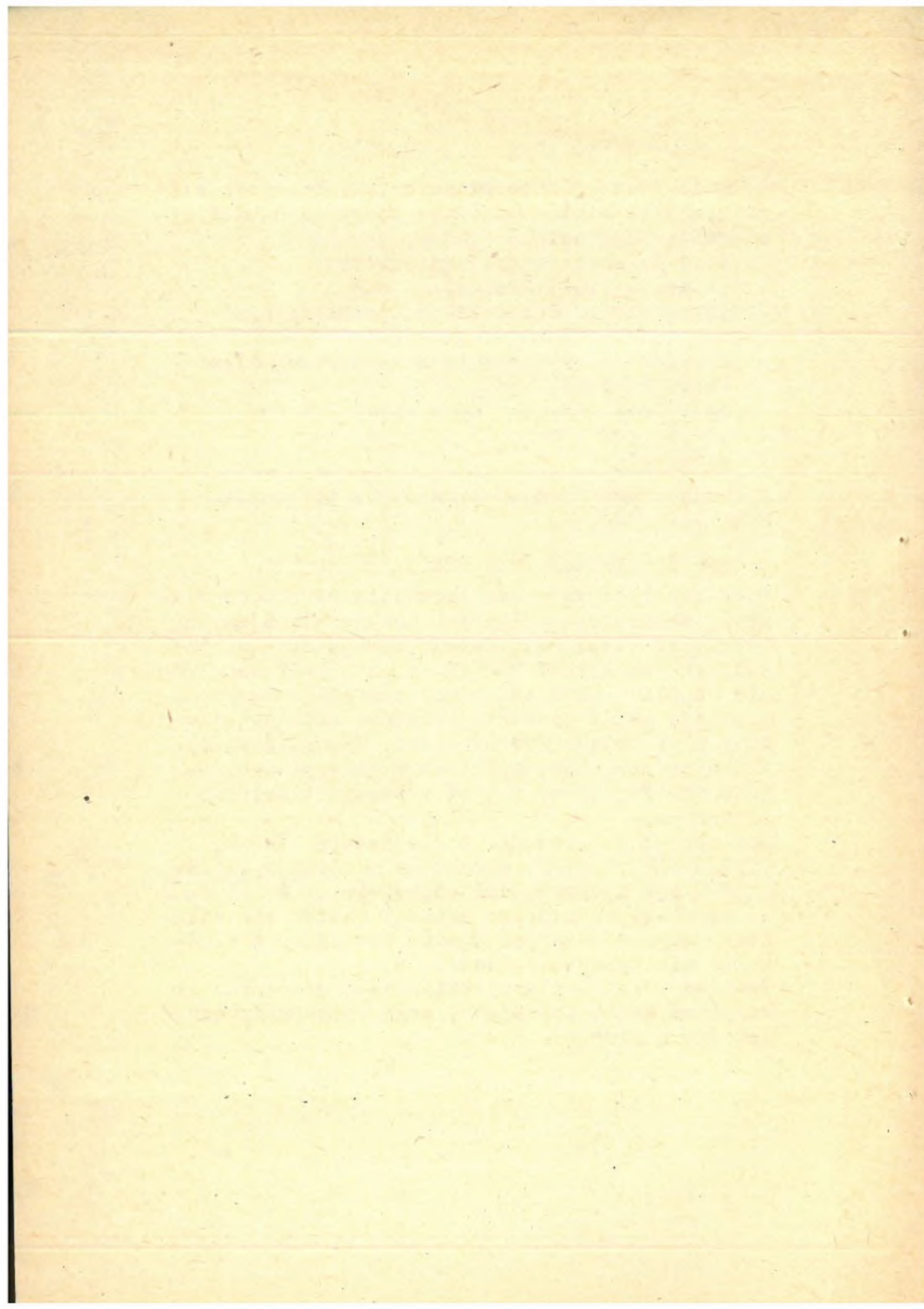
Rysunek 1 przedstawia klawiaturę terminala MERA 7953 VGD.

1.1. Znaki specjalne i znaki sterujące

Operator steruje pracą terminala za pomocą klawiszy znaków specjalnych i znaków sterujących. Odnośnymi klawiszami znaków specjalnych są: TAB, ESC, CR /względnie RETURN/ oraz RUBOUT /względnie DELETE/. Znak sterujący zostaje wytworzony wówczas, kiedy operator naciśnie klawisz literowy przy wciśniętym klawiszu CTRL /sterujący/. W tym podręczniku, znaki sterujące pisane są jako CTRL/x, gdzie x jest wybranym klawiszem literowym.

Gdy operator wprowadzi z klawiatury CTRL/R, CTRL/U bądź CTRL/Z, egzekutor realizuje odpowiednią funkcję i wyprowadzi odpowiednio \hat{R} , \hat{U} i \hat{Z} na terminal, z którego wysłany został ten znak sterujący. Dla innych znaków sterujących egzekutor nie wprowadza znaku \hat{x} .

Tablica 1 zawiera wszystkie znaki specjalne i dostępne znaki sterujące, oraz opisuje funkcje każdego z nich.



TABLICA 1

Znaki specjalne i znaki sterujące

- CR - CR /powrót karetki/ jest znakiem, który kończy wiersz wprowadzanych informacji i przemieszcza karetkę względnie kursor na pozycję 1 w następnym wierszu.
- ESC - Znak ESC kończy wiersz wprowadzanych informacji bez przemieszczania karetki czy kursora. Gdy znak ten zostanie użyty do zakończenia komendy MCR, eliminuje on domniemane przynaglenie MCR. W systemach, które mają wygenerowaną odpowiednio opcjonalną możliwość, klawisz ESC może być pierwszym znakiem trybu ESCAPE (patrz "Opis Drajwerów Urzędzeń Zewnętrznych" oraz opis komendy SET /ESCSEQ).
- DELETE - Klawisz DELETE usuwa ostatni znak napisany na terminalu oraz poprzednie kolejne znaki, jeśli będzie naciskany wielokrotnie. Pierwsze naciśnięcie DEL usuwa znak i wyprowadza znak (\), po którym następuje wyprowadzenie usuniętego znaku. Dalsze usunięcia powodują wyprowadzenie na terminalu usuniętych znaków. Następnie, gdy operator wprowadzi pierwszy znak nie będący DEL, system drukuje kończący znak (\), a po nim wprowadzany znak.
- Na przykład:
---pierwszy DEL
!---drugi DEL
! !---trzeci DEL
! ! !---pierwszy znak różny od DEL
MISTKAE\EAK\AKE
wynikiem jest MISTAKE
- Na terminalu ekranowym, który posiada cofanie /opcja generacji systemu/ i został zadeklarowany jako będący typu

CRT za pomocą komendy SET /CRT=TTN:, klawisz DELETE przemieszcza kursor o jedną pozycję w lewo i usuwa każdy znak wyświetlany na tej pozycji. Na przykład, celem skorygowania słowa MISTKAE, należy wcisnąć DELETE trzykrotnie, celem usunięcia kolejno liter E,A,K. W wyniku tego otrzymujemy MIST, a następnie piszemy AKE, celem otrzymania MISTAKE.

CTRL/C - CTRL/C, naciśnięte jako pierwszy znak w wierszu, albo gdy terminal wysła dane, powoduje, że MCR zgłasza się za pomocą jawnego przynaglenia (MCR>). (Dwa wyjątkowe skutki naciśnięcia CTRL/C są opisane poniżej).

Jeśli zadanie oczekuje na wejście, kiedy operator nacisnął CTRL/C, a potem znak <CR>, to MCR wyprowadza przynaglenie, a następnie zwraca sterowanie przerwaniem zadaniu po otrzymaniu jednego wiersza informacji wejściowej. /Wyjątek ma miejsce wtedy, gdy przywiązane zadanie specyfikuje asynchroniczną pułapkę systemową (AST) dla niespodziewanych znaków. W tym przypadku CTRL/C powoduje pojawienie się AST, przez co CTRL/C wywołuje procedurę obsługi AST w programie a nie MCR/. Dokładny opis stosowania CTRL/C podane jest w pkt.

Skutki dodatkowe:

- Jeśli znakiem wprowadzonym jako ostatni z terminala, było CTRL/S, wówczas CTRL/C realizuje dodatkowo funkcje CTRL/Q.
- Jeśli terminal został ustawiony w tryb stronicowania wyprowadzenia mianowicie za pomocą opcji /HOLD komendy SET MCR, CTRL/C ma skutek

taki, jak komenda SET /NOHOLD. Innymi słowy, CTRL/C anuluje tryb stronicowania.

- CTRL/I - CTRL/I /względnie TAB/ przemieszcza bieżącą pozycję drukowania do następnej poziomej tabulacji w danym wierszu. TAB System ustanawia tabulacje w każdej, co-ósmej pozycji w wierszu.
- CTRL/K - CTRL/K powoduje tabulację pionową, realizując cztery wysuw wiersza.
- CTRL/L - CTRL/L powoduje wpisanie znaku nowej strony /lecz nie powoduje stronicowania/. Naciśnięcie go powoduje na terminalu 8 wysuwów wiersza.
- CTRL/O - CTRL/O naprzemiennie blokuje i podejmuje wyprowadzanie informacji na terminal. Wynik naciśnięcia CTRL/O zależy od stanu terminala w chwili, gdy operator nacisnął ten znak. System steruje funkcją CTRL/O wewnętrznie, za pomocą pojedynczego bitu związanego z terminalem nazywanego bitem blokowania wyprowadzenia. Dla nieprzywiązanych terminali, bit ten jest także zerowany za każdym razem, gdy zadanie inicjuje zlecenie wyprowadzenia informacji na terminal. Uwzględniając te warunki zerowania bitu blokowania oraz, że naciśnięcie CTRL/O zawsze neguje ten bit, CTRL/O wpływa na terminal, który go wysłał w następujący sposób:
- Na przywiązonym terminalu operator może zatrzymać wyprowadzenie informacji przez naciśnięcie CTRL/O. Informacje wyjściowe nie będą pojawiać się na terminalu, aż do następnego:
- odwołania terminala
 - nadejścia spodziewanych informacji wejściowych

- nadejścia niespodziewanych informacji wejściowych,

- CTRL/O.

Na nieprzywiązanych terminalach CTRL/O zatrzymuje wyprowadzenie tylko dla bieżącego bufora, gdyż przy następnej inicjacji operacji wejścia/wyjścia, system zawsze zeruje bit blokowania wyprowadzania.

CTRL/Q - CTRL/Q naciśnięcie po CTRL/S podejmuje wyprowadzenie informacji, zawieszona przez uprzednie CTRL/S. /Cecha ta jest opcją generacji systemu/.

CTRL/R - Naciśnięcie CTRL/R przed naciśnięciem ogranicznika wiersza powoduje, że system ponownie wyprowadza bieżący wiersz w nowym wierszu, pomijając wszelkie usunięte znaki. Można wtedy kontynuować wprowadzanie danego wiersza. Jeśli bieżący wiersz jest pusty, CTRL/R realizuje powrót karetki i zmianę wiersza.

Na przykład:

```
MISTKAE\EAK\AKE<CTRL/R>
```

```
MISTAKE
```

CTRL/S - Naciśnięcie CTRL/S w czasie, gdy terminal otrzymuje informacje wyjściowe zawieszają wyprowadzanie informacji aż do chwili, gdy operator naciśnie CTRL/Q bądź CTRL/C. Zawieszane informacje zostają jedynie opóźnione, a nie utracone /patrz opis CTRL/O/. Połączone funkcje CTRL/Q i CTRL/S są dogodnie przy korzystaniu z terminala CRT /Cecha ta jest opcją generacji systemu/.

CTRL/U - Naciśnięcie CTRL/U przed naciśnięciem ogranicznika wiersza powoduje usunięcie napisanego wiersza. System reaguje powrotem karetki i zmianą wiersza,

tak iż wiersz można napisać na nowo. Po naciśnięciu CTRL/U system odpowiada wyprowadzeniem ^U na terminal.

CTRL/X - Naciśnięcie CTRL/X powoduje zerowanie bufora wprowadzania z wyprzedzeniem. Bufor przechowuje znaki wejściowe do chwili, gdy drajwer terminala będzie mógł je przyjąć. Kiedy jest on pełny znaki wejściowe nie są przyjmowane. Naciśnięcie CTRL/X zeruje bufor.

CTRL/Z - CTRL/Z jest znakiem, oznaczającym koniec zbioru. Wykorzystywany jest on jako sygnał dla zadań systemowych, takich jak MACRO, PIP, TKB, celem zasygnalizowania, że użytkownik skończył korzystanie z nich i zadanie systemowe może się zakończyć. Po naciśnięciu CTRL/Z system odpowiada wyprowadzeniem ^Z na terminal.

1.2. Parametry terminala

W systemie operacyjnym DOC PB może równocześnie pracować kilka terminali. Każdy terminal pracuje niezależnie od innych tak, że z każdego może być uruchomione jedno lub więcej różnych zadań. W ten sposób można z różnych terminali uruchomić równocześnie na przykład program użytkowy PIP i assembler MACRO pod warunkiem, że jest dostępna wymagana wielkość pamięci.

1.2.1. Uprzywilejowanie terminala

Operator może wydawać komendy uprzywilejowane tylko z uprzywilejowanego terminala. /W opisie poszczególnych komend MCR podano, które z tych komend są uprzywilejowane/. W systemach DOC PB, które używają zabezpieczenia wieloużytkownikowego, poszczególni użytkownicy są albo uprzywilejowani albo nieuprzywilejowani. Gdy użytkownik rozpoczyna pracę z systemem, terminalom nadawany jest stan uprzywilejowania na podsta-

wie kodu użytkownika. W innych systemach DOC PB stan uprzywilejowania terminala wyznaczany jest początkowo przy generacji systemu. Operator może używając komendy MCR SET z uprzywilejowanego terminala zmodyfikować stan uprzywilejowania każdego innego terminala przyłączonego do systemu.

UWAGA

Gdy jest więcej niż jeden terminal uprzywilejowany operatorzy muszą postępować szczególnie ostrożnie. Uprzywilejowane komendy mogą zakłócać się wzajemnie, a także zakłócać pracę systemu.

1.2.2. Terminal przywiązany i nieprzywiązany

Terminal jest przywiązany wówczas, gdy wszystkie informacje wprowadzone i wyprowadzane kierowane są do /względnie z/ zadania innego niż MCR. Na przykład, gdy programista wywołuje zadanie /tak jak na przykład edytor EDI/, zadanie zwykle przywiązuje się do terminala tak, że żadne inne zadanie nie może użyć tego terminala dla wykonania operacji wejścia/wyjścia. Gdy programista zakończy pracę z zadaniem /na przykład naciśnięszy CTRL/Z/, zadanie odwiązuje się samo od terminala, który kieruje dalsze wprowadzane informacje do MCR.

Ze względu na to, że przywiązany terminal kieruje wszystkie wprowadzane informacje do zadania, do którego jest przywiązany, operatorem skomunikowania się z MCR musi albo wyjść z zadania, albo wywołać MCR.

1.2.3. Terminal podporządkowany

Można udostępnić terminal wyłącznie dla jednego lub kilku zadań, poprzez wykonanie komendy MCR SET, która ustawia terminal w stan podporządkowania. Różnica pomiędzy terminalem podporządkowanym a terminalem przywiązanym polega na tym, że terminal podporządkowany odrzuca

wszelkie niespodziewane informacje wejściowe, w tym CTRL/C, z wyjątkiem CTRL/O, CTRL/Q i CTRL/S. Natomiast przywiązane terminale przyjmują pewne typy niespodziewanych informacji wejściowych. Aż do chwili, gdy operator wyda następną komendę MCR /lub zadanie wyda następną specjalną operację wejścia/wyjścia celem usunięcia stanu podporządkowania terminal może być wykorzystywany tylko do komunikowania się z zadaniem za pomocą spodziewanej informacji wejściowej. Terminale podporządkowane są używane zwykle w zastosowaniach przetwarzania w czasie rzeczywistym.

1.3. Przynaglenia wejścia

Istnieją trzy typy przynaglenia, które wskazują na to, że terminal oczekuje na wprowadzenie informacji:

1. Domniemane przynaglenie (>),
2. Przynaglenie od zadania <tsk>),
3. Przynaglenie od MCR (MCR>).

1.3.1. Przynaglenie domniemane

Przynaglenie domniemane (znak > na pozycji 1 nowego wiersza) sygnalizuje, że terminal jest gotowy do przyjęcia niespodziewanej informacji wejściowej. Pojęcie "niespodziewana" oznacza, że żadne specyficzne zadanie nie zleciło wprowadzania informacji z tego terminala. Informacja wprowadzona przez operatora w odpowiedzi na domniemane przynaglenie, kierowana jest zwykle do MCR. Gdy MCR otrzymuje niespodziewane informacje wejściowe, wówczas identyfikuje, analizuje i odpowiada na nie odpowiednio do treści.

1.3.2. Przynaglenie od zadania

Gdy MCR aktywuje zadanie do obsłużenia komendy, zadanie zwykle przynagla <tsk>

gdzie tsk jest 3-znakową nazwą zadania. /Wszystkie zadania systemowe identyfikują się same

3-znakowym przynagleniem, i również zadania użytkownika powinny robić tak samo/.
W czasie, gdy przynaglające zadanie jest przywiązane do terminala, zadanie to otrzyma wszystkie spodziewane informacje wejściowe, wprowadzone z tego terminala, dopóki się nie odwiąże. /Terminal pomija wszystkie niespodziewane informacje wejściowe/. Natomiast zadaniu nieprzywiązanemu może przydarzyć się, że inne zadanie będzie spodziewało się wprowadzenia informacji z danego terminala zanim operator wprowadzi odpowiedź na przynaglenie. Zakłócające zadanie otrzyma wówczas informacje, które miały być wprowadzone do zadania, które wyprowadziło przynaglenie. Aby uniknąć tej sytuacji, każde zadanie, które spodziewa się wprowadzenia informacji z terminala winno wykonać następujące operacje:

- przywiązać się do terminala,
- przynaglić odpowiednio uformatowanym identyfikatorem (to jest tsk>), a następnie
- oczekiwać na wprowadzenie informacji /wydając zlecenie czytania/.

1.3.3. Przynaglenie od procedury MCR

Informacje wysłane z terminala w odpowiedzi na jawne przynaglenie:

MCR>

zostaną zawsze skierowane do MCR.

Można wywołać jawne przynaglenie MCR wysyłając:

CTRL/C

bądź /jak pokazano w drugim przypadku opisanym poniżej/:

CTRL/C < CR >

CTRL/C nie wywołuje przynaglenia MCR w wyjątkowym przypadku, opisanego w drugiej uwadze poniżej. Ponadto, opis CTRL/C w tablicy 1 podaje dwie dalsze sytuacje, w których CTRL/C nie spowoduje wysłania przynaglenia przez MCR.

Poniższe uwagi wyjaśniają rozmaite zastosowania jawnego przynaglenia MCR:

- Gdy terminal wyprowadza domniemane przynaglenie /patrz punkt 1.3.1./, nie ma gwarancji, że informacje wysłane w odpowiedzi zostaną skierowane do MCR. Aby zapewnić, że MCR otrzyma wprowadzone informacje, operator może wcisnąć CTRL/C celem wywołania przynaglenia MCR. Gdy MCR przetworzy pojedynczy wiersz wprowadzanych informacji, terminal wyprowadzi domniemane przynaglenie ').
- Gdy terminal wyprowadza przynaglenie od zadania (tsk>), wówczas użytkownik może wywołać MCR, naciskając:

CTRL/C <CR>

Gdy MCR przetworzy otrzymany pojedynczy wiersz wprowadzanych informacji, wówczas terminal ponownie wyświetli przynaglenie pierwotnego zadania *).

UWAGA

Jeśli przywiązane zadanie specyfikuje AST po otrzymaniu niespodziewanego wprowadzania informacji, CTRL/C powoduje wystąpienie AST, przez co nastąpi wywołanie procedury obsługi AST w zadaniu a nie procedury MCR. W konsekwencji nie pojawi się przynaglenie od MCR.

- W czasie, gdy jakkolwiek terminal otrzymuje informacje wyjściowe z zadania, operator może przerwać wprowadzanie informacji naciskając CTRL/C. Po zakończeniu bieżącej informacji wejścia/wyjścia, terminal wyprowadza jawne przynaglenie MCR. Gdy MCR przetworzy wysłany doń pojedynczy wiersz wprowadzanych informacji, to system podejmuje przerwane wprowadzanie informacji *).

*) Pojedynczy wiersz wprowadzanych informacji otrzymywany przez MCR, może w rzeczywistości wywołać zadanie, które wysła swoje własne przynaglenie. Na przynaglenia z wywołanego zadania może operator odpowiedzieć przez naciśnięcie CTRL/C, celem wywołania przynaglenia MCR. Operator może wówczas zlecić inne zadanie, które wysła swoje własne przynaglenie, itd.

2. Urządzenia

Zarządzanie wszystkimi przyłączonymi urządzeniami peryferyjnymi jest ważną funkcją dla zapewnienia wydajnego działania całego systemu. Punkt ten podaje podstawowe informacje niezbędne dla zrealizowania tej funkcji.

Każde urządzenie ma unikalny identyfikator, złożony z dwóch znaków ASCII tworzących nazwę urządzenia, oraz jedno- lub dwucyfrowego oktalnego numeru jednostki, po których następuje dwukropek /na przykład, DK1:, MT2:/. Jeśli numer jednostki zostanie pominięty, to system przyjmuje z domniemania, że chodzi o jednostkę zero, a zatem LP: wskazuje na drukarkę wierszową o numerze 0.

Pseudourządzenia muszą zawsze odpowiadać urządzeniom fizycznym. Wzajemne połączenie deklarowane jest poprzez komendę przypisania ASN lub przedadresowania RED. Pseudourządzenia pozwalają w prosty sposób odwoływać się do urządzeń fizycznych /np. pseudourządzenie SY: odpowiada dyskowi systemowemu/, co upraszcza strukturę domniemań.

Tablica 2 wymienia nazwy urządzeń peryferyjnych oraz nazwy odpowiadające tym urządzeniom, współpracującym z systemem DOC PB.

TABLICA 2
URZĄDZENIA PERYFERYJNE SYSTEMU OPERACYJNEGO
DOC PB

NAZWA URZĄDZENIA	IDENTYFIKATOR URZĄDZENIA
CZYTNIK KART (SMCR)	CRnn:
TAŚMOWA PAMIĘĆ KASETOWA (SMTA)	CTnn:
PAMIĘCI DYSKOWE	
(SMRP4/5/6)	DBnn:
(SMRF)	DFnn:
(SMRK)	DKnn:
(SMRK6/7)	DMnn:

(SMRL1/2)	DLn:
(SMRM2/3)	DRn:
(SMRP2/3)	DPn:
(SMRS3/4)	DSn:
(SMRX1)	DXn:
(SMRX2)	DYn:
PODSYSTEM GRAFICZNY (SMVT)	GRn:
DRUKARKA WIERSZOWA (SMLP)	LPn:
PAMIĘĆ TAŚMOWA (SMTU45/16/77, SMTE16)	MMn:
(SMTU10, SMTS3)	MTn:
(SMTS4)	MSn:
PERFORATOR TAŚMY PAPIEROWEJ (SMPC)	PPn:
CZYTNIK TAŚMY PAPIEROWEJ (SMPC)	PRn:
TERMINAL (SMDL/SMDH/SMDJ/SMDZ)	TTn:
RÓWNOLEGŁY INTERFEJS LINIOWY (SMDA-B)	XBn:
ASYNCHRONICZNY INTERFEJS LINIOWY (SMDL-E)	XLn:
INTERFEJS MIĘDZYPROCESOROWY (SMDMC)	XMn:
SYNCHRONICZNY INTERFEJS LINIOWY (SMDP)	XPn:
	(SMDQ) XQn:
	(SMDUP) XWn:
	(SMDU) XU n:
SYSTEM PAMIAROWY CAMAC (SMCC-CA106)	CCn:

TABLICA 3

PSEUDOURZĄDZENIA

PSEUDOURZĄDZENIA	NAZWA
KONSOLOWE URZĄDZENIE LISTOWANIA	CL:
KONSOLOWE URZĄDZENIE WYJŚCIOWE	CO:
PSEUDOTERMINAL	TI:
DOMNIEMANE URZĄDZENIE SYSTEMOWE	LB:
DOMNIEMANE URZĄDZENIE UŻYTKOWNIKA	SY:
URZĄDZENIE ZEROWE	NL:

Zadanie realizuje operacje wejścia/wyjścia do jednostek logicznych, identyfikowanych numerami, nazywanymi numerami jednostek logicznych (LUN). Numery (LUN) przypisywane są poszczególnym urządzeniom przez programistę bądź operato-

ra zanim program użyje numeru LUN. Przypisanie LUN następuje w jednym z trzech etapów:

- Przy budowie zadania
- Z zadania, w czasie jego wykonywania, za pośrednictwem dyrektywy egzekutora,
- Za pośrednictwem komendy zmiany przypisania MCR (REASSIGN), po zainstalowaniu zadania, w czasie gdy jest ono uśpione.

Bliższe informacje o jednostkach logicznych czytelnik znajdzie w podręczniku: "Drajwery urządzeń wejścia/wyjścia", bądź w podręczniku programisty dla odnośnego języka.

2.1. Pseudourządzenia

Pseudourządzenie jest nazwą jednostki urządzenia, które nie jest rzeczywistym urządzeniem, aż do chwili, gdy nastąpiło jego przeadresowanie. Każde zadanie w systemie, w tym także systemowe, wymaga komunikowania się z jednym lub więcej z następujących pseudourządzeń:

CO: konsolowe urządzenie wyjściowe

CL: konsolowe urządzenie listowania

TI: pseudoterminale

LB: domniemane urządzenie systemowe

SY: urządzenie systemowe użytkownika

W danym zadaniu, LUN1 może być przypisany do CL:. Gdy zadanie użyje LUN1, to dane zostaną wysłane do tego urządzenia, do którego operator przeadresował urządzenie CL:. Operator może przeadresować CL: do jakiegokolwiek odpowiedniego urządzenia /na przykład drukarki wierszowej lub terminala/ nie wpływając na przebieg wykonywania zadania.

Pseudourządzenie przyjmuje stan uprzywilejowania urządzenia fizycznego, do którego zostało przeadresowane. Jeśli TT: jest terminalem uprzywilejowanym, to CL: staje się uprzywilejowanym w momencie, gdy zostało przeadresowane do tego terminala. Jednakże, jeśli CL: zostaje następnie przeadresowane do nieuprzywilejowanego terminala to również stanie się nieuprzywilejowanym.

Zwykle preadresowuje się CO: do terminala słownego operatora, zaś CL: do drukarki wierszowej. System automatycznie preadresowuje TI:, najpowszechniej stosowane pseudourządzenie, zależnie od tego, jak zadanie zostało aktywowane.

- Jeśli operator wydał komendę RUN MCR celem aktywowania zadania, TI: dla danego zadania zostało preadresowane do terminala, który wydał komendę
- Jeśli jakieś inne zadanie zostanie aktywowane przez dyrektywę egzekutora RUN α , to TI: zadania zostaje preadresowane przez domniemanie do TT: zadania, które wydało dyrektywę RUN α .
- Jeśli egzekutor umieścił zadanie w kolejce zegara dla aktywacji po wyspecyfikowanym odstępie czasu, TI: zadania zostanie preadresowane przez domniemanie na CO: /komenda RUN MCR oraz dyrektywa RUN α egzekutora pozwalają użytkownikowi specyfikować, że zadanie ma być aktywne po wyspecyfikowanym odstępie czasu od wydania komendy względnie dyrektywy/.

2.2. Urządzenia zerowe

Testowanie programu wymaga często "źródła danych" i "upustu danych". Źródło danych jest urządzeniem, stosowanym jako źródło nieograniczonych danych; zaś upust danych jest urządzeniem zdolnym do przyjęcia nieograniczonej liczby danych. Urządzenia te są tylko mechanizmami testowania, a nie urządzeniami rzeczywistymi. Testowane programy wykorzystują "urządzenie zerowe" jako służące zarówno jako źródło danych, jak i upust danych. Na przykład, przy testowaniu programu, który normalnie generuje duże liczby wydruków, użytkownik może preadresować względnie przypisać odpowiedni LUN do NL:. System pomija wówczas wyprowadzane informacje, skierowane do NL:. Gdy program czyta z NL:, urządzenie zerowe odpowiada kodem końca zbioru (IE.EOF); gdy program pisze do NL:, urządzenie

zerowe odpowiada kodem wskazującym poprawne zakończenie operacji (IE.SUC).

2.3. Urządzenia logiczne

Nazwy urządzeń logicznych stanowią kolejny sposób, za pomocą którego zadania mogą być niezależne od urządzeń. /Cecha ta jest opcją generacji systemu/. Nazwa urządzenia logicznego ma tą samą składnię, co rzeczywista jednostka urządzenia; składa się z nazwy /dwóch liter ASCII/, opcjonalnego 1-lub 2-cyfrowego oktalnego numeru jednostki, zakończonych znakiem dwukropka. 2-znakowa nazwa może być albo równoważnikiem standardowej nazwy urządzenia systemu operacyjnego DOC PB /na przykład DK:/, lub może składać się z dwóch liter wybranych dowolnie /na przykład XY:/. Zanim zadanie odwoła się do urządzenia logicznego zamiast do rzeczywistej jednostki urządzenia, operator musi wydać komendę przypisywania (ASN) MCR, celem związania nazwy logicznego urządzenia zadania, z rzeczywistą jednostką urządzenia.

Istnieją trzy rodzaje przypisań urządzeń logicznych: globalne, lokalne i na czas pracy użytkownika z systemem.

- Przypisanie globalne dotyczy wszystkich zadań wykonywanych w systemie.
- Lokalne przypisanie urządzeń logicznych dotyczy tylko zadań zainicjowanych z danego terminala, użytego dla dokonania przypisań. Przypisania lokalne są nadrzędne wobec sprzecznych z nimi przypisaniami globalnymi bądź przypisaniami na czas pracy użytkownika z systemem. Ponadto, ta sama nazwa urządzenia logicznego może być lokalnie przypisana innemu urządzeniu przez inny terminal.
- Przypisania urządzenia logicznego, dokonywane na czas pracy użytkownika z systemem, dotyczą systemów, które używają zabezpieczenia wieloużytkownika. Gdy użytkownik wyda ko-

mendę MCR HELLO, celem rozpoczęcia pracy z systemem, system automatycznie wykonuje jedno lub więcej przypisań urządzeń logicznych. Takie przypisania są nadrzędne wobec przypisań globalnych, natomiast przypisania lokalne wykonane z danego terminala, są nadrzędne zarówno nad przypisaniami globalnymi, jak i przypisaniami na czas pracy użytkownika z systemem.

W sytuacji, w której lokalne przypisanie jest nadrzędne wobec przypisania ustanowionego na czas pracy użytkownika z systemem, system przywraca przypisanie utworzone w czasie rozpoczęcia pracy użytkownika z systemem z chwilą, gdy użytkownik anuluje lokalne przypisanie nazwy urządzenia.

Przypisanie urządzeń logicznych na czas pracy użytkownika z systemem obowiązują aż użytkownik zakończy pracę lub do chwili, gdy uprzywilejowany terminal anuluje przypisanie. Egzekutor tworzy tablice urządzeń logicznych, w której zapisane są wszystkie przypisania urządzeń logicznych i zaznaczony jest w niej rodzaj przypisania /lokalne, na czas pracy użytkownika bądź globalne/. Egzekutor następnie przeszukuje te tablice za każdym razem, gdy zadanie lub komenda odwoła się do jakiegoś urządzenia. Nazwa logiczna znaleziona w tablicy jest nadrzędna nad fizyczną jednostką urządzenia, mająca ten sam identyfikator.

3. Funkcje zabezpieczenia wieloużytkownikowego

Zabezpieczenie wieloużytkownikowe, będące opcją generacji systemu, pozwala instalacji systemu operacyjnego DOC PB nadzorować prace poszczególnych użytkowników systemu.

3.1. Rozpoczynanie i kończenie pracy z systemem

Po to, aby system pozwolił użytkownikowi wydać komendy MCR bądź uruchomić zadanie, poza ko-

meną HELP MCR, użytkownik musi najpierw rozpocząć pracę z systemem wydając komendę HELLO. Zależnie od UIC użytkownika, terminal staje się wówczas uprzywilejowany bądź nieuprzywilejowany. Po zakończeniu pracy, każdy użytkownik kończy pracę z systemem wydając komendę BYE. Proces rozpoczynania i kończenia pracy z systemem pozwala prowadzić ewidencję kont wszystkich użytkowników.

Po rozpoczęciu lub zakończeniu pracy z systemem na terminalu wyprowadzane są następujące informacje:

```
time LOGIN USER lastname [g,m] TTn:  
time LOGOUT USER [g,m] TTn:
```

3.1.1. Komenda HELLO

Parametry komendy HELLO składają się z UIC użytkownika /względnie jego nazwiska/ oraz hasła. Standardowym formatem dla specyfikacji UIC jest [g,m], gdzie g i m są liczbami oktalnymi od 1 do 377, które reprezentują odpowiednio numer grupowy i numer użytkownika. Komenda HELLO pozwala na stosowanie różnych formatów UIC. Wyszpecyfikowany format UIC wpływa na to, jak system odpowie na tą komendę. Uprzywilejowany użytkownik ma numer grupowy mniejszy lub równy 10. Każdy kod UIC ma związany z nim ciąg od 1 do 6 znaków, czyli hasło, które chroni przed dostępem do systemu przez osoby nieupoważnione. Nie podając ważnego hasła dla wyspecyfikowanego kodu UIC /względnie nazwiska/, nikt nie może przyłączyć się do systemu.

Na przykład:

```
HELLO <CR>
```

```
/Dzień dobry, powrót karetki/
```

```
ACCOUNT OR NAME: [307,317]<CR>
```

```
/Konto lub nazwa/
```

```
PASSWORD: KAROL
```

```
/hasło: KAROL/
```

Należy zwrócić uwagę na to, że ten przykład komendy HELLO specyfikuje nazwisko użytkownika, na podstawie której system określi jego kod UIC. Gdy użytkownik rozpoczął pracę z systemem, system automatycznie przypisuje nazwę logiczną SY: do pamięci dyskowej użytkownika /przypisanie urządzenia logicznego przy rozpoczęciu pracy przez użytkownika/ i przeszukuje odpowiednie UFD w celu znalezienia zbioru LOGIN.CMD. Jeśli znajdzie, to zbiór jest przesyłany do procesora zbioru komend /patrz podręcznik "Zbiory komend"/. Jest to szczególnie użyteczne przy konieczności wydawania tych samych komend przy każdym rozpoczęciu pracy z systemem. Na przykład, jeśli użytkownik utworzył zbiór LOGIN.CMD zawierający

```
SET /LOWER=TI:  
SET /BUF=TI:80  
SET /CRT=TI:
```

następujące informacje zostają wyświetlone przy rozpoczęciu pracy z systemem:

```
>HEL NOWAK  
PASSWORD:  
DOC PB-2 BL26 MULTI-USER SYSTEM  
GOOD AFTERNOON  
11-FEB-83 24:06 LOGGED ON TERMINAL TT20:  
  
>@LOGIN.CMD  
>SET /LOWER=TI:  
>SET /BUF=TI:80  
>SET /CRT=TI:  
>@(<EOF)
```

Gdy numer grupowy jest 11 lub większy, terminal jest nieuprzywilejowany /patrz punkt 1.2.1./.

3.1.2. Komenda BYE

Użytkownik, który zakończył swą pracę, wydaje komendę BYE (do widzenia) celem zakończenia pracy z systemem. Komenda ta nie ma parametrów. Po prostu informuje ona system, że użytkownik zakończył pracę. System usuwa wówczas wszystkie

przypisania lokalne i utworzone przy rozpoczęciu pracy z systemem, demontuje wszystkie zamontowane urządzenia prywatne /patrz punkt 3.2. poniżej/, zwalnia wszystkie przydzielone urządzenia prywatne, usuwa wszystkie aktywne nieuprzywilejowane zadania, które zlecono z tego terminala, wyświetla komunikat powiadamiający o zakończeniu i zapobiega dalszemu wykorzystaniu terminala aż do chwili rozpoczęcia pracy przez innego użytkownika.

Przykład:

>BYE

>

HEVE A GOOD AFTERNOON

15-DEC-82 2:15 TT5: LOGGED OFF

3.2. Urządzenia ogólnodostępne, urządzenia prywatne i niczyje

System, który posiada zabezpieczenie wieloużytkownikowe, nadaje urządzeniom cechy specjalne. Urządzenia mogą być albo ogólnodostępne albo prywatne albo niczyje, czyli nie być ani ogólnodostępnymi ani prywatnymi. Cechy te definiowane są w następujący sposób:

- Urządzenie ogólnodostępne jest urządzeniem, z którego może korzystać każdy, kto przyłączył się do systemu. Użytkownicy uprzywilejowani mogą uczynić urządzenie ogólnodostępnym przez użycie odpowiednich kluczy przy komendzie SET. Jeśli uprzywilejowany użytkownik montuje tom na urządzeniu ogólnodostępnym, to wszyscy użytkownicy systemu mają dostęp do zbiorów na tym tomie. Jednakże, w następstwie wydania komendy demontującej wstrzymany jest dostęp do zbiorów, a każdy użytkownik chcąc mieć dostęp do tomu musi wydać komendę MOU. Urządzenia ogólnodostępne nie mogą być przydzielane.
- Urządzenie prywatne jest urządzeniem, które użytkownik przydzielił /za pomocą komendy

przydzielania ALL/. Tylko uprzywilejowani użytkownicy i właściciele urządzeń prywatnych mają dostęp do urządzeń w celu użycia ich w systemie zbiorów DOC PB /patrz punkt 3.1./. Jeśli nieuprzywilejowany użytkownik przydziela urządzenie i montuje na min tom, to inni nieuprzywilejowani użytkownicy nie mają dostępu do tomu, uprzywilejowani użytkownicy mają dostęp do tomu po wydaniu komendy MOU.

- Urządzeniem niczym jest urządzenie, które nie jest ani ogólnodostępnym ani prywatnym. Jeśli użytkownik, zarówno uprzywilejowany jak i nieuprzywilejowany, montuje tom na niczym urządzeniu, wszyscy użytkownicy systemu mogą, po wydaniu komendy MOU, otrzymać dostęp do tomu. Żaden użytkownik nie ma dostępu do tomu przed wydaniem komendy MOU. Nieuprzywilejowani użytkownicy mogą przydzielać lub montować tomy tylko na urządzeniach niczych. Uprzywilejowani użytkownicy mogą przydzielać i tworzyć tomy ogólnodostępne lub montować tomy na urządzeniach niczych.

Użytkownik uprzywilejowany może określić jakiekolwiek urządzenie w systemie jako urządzenie ogólnodostępne. Urządzeniami najpowszechniej używanymi jako "ogólnodostępne" są: drukarka wierszowa i dyskowa pamięć systemowa. Komenda SET z kluczem /PUB określa urządzenie jako ogólnodostępne.

Komenda MCR ALL (przydziel) czyni wyspecyfikowane urządzenie niczyje własnością użytkownika, który wydał tę komendę. Przydzielenie urządzenia zapobiega, aby inny nieuprzywilejowany użytkownik nie uzyskał dostępu do tego urządzenia. /Użytkownicy uprzywilejowani mogą dokonać nadrzędnego przydziału wobec poprzedniego przydziału przez nieuprzywilejowanego użytkownika/. Gdy użytkownik sięga do urządzenia, bez przydzielenia go, inny użytkownik może uzyskać dostęp i/lub przydzielić to urządzenie.

Komenda usunięcia przydziału (DEA), zmienia prywatne urządzenie z powrotem na urządzenie niczyje, pozwalając w ten sposób innym użytkownikom uzyskać dostęp do niego. Użytkownicy uprzywilejowani mogą usuwać przydział jakiego - kolwiek urządzenia prywatnego, bez względu na to, kto jest jego właścicielem. Użytkownicy nieuprzywilejowani mogą usuwać przydział tylko swoich własnych urządzeń prywatnych. Gdy użytkownik odłącza się od systemu, system automa - tycznie demontuje i usuwa przydziały jakichkol - wiek urządzeń prywatnych, należących do tego użytkownika. W celu efektywnego wykorzystania zasobów systemu, powinna obowiązywać zasada, by użytkownicy usuwali przydział urządzeń, gdy tyl ko urządzenia te nie są już im potrzebne. Niektóre programy systemu DOC PB takie jak INI, DSC, BAD i FLX mają dostęp do niezamontowanych tomów.

3.3. Program aktualizacji katalogu użytkowników systemu

System operacyjny DOC PB zawiera program zarzą - dzania zbiorem katalogowym użytkowników systemu (ACNT) dla tworzenia i modyfikowania tego zbioru. Za każdym razem, gdy użytkownik próbuje roz - począć pracę z systemem z terminala, system bada parametry komendy HELLO przy użyciu katalogu użytkowników systemu, celem stwierdzenia, czy użytkownik może uzyskać dostęp do systemu czy też nie /patrz punkt 3.1./. Katalog użytkowni - ków systemu opisuje wszystkie kody UIC, które zostały utworzone i mogą być użyte w systemie z zabezpieczeniem wieloużytkownikowym. Jeden kod UIC może mieć szereg użytkowników, z któ - rych każdy ma odrębne hasło.

Program ACNT jest programem konwersacyjnym, któ - ry pozwala uprzywilejowanym użytkownikom:

- Tworzyć zbiór użytkowników systemu (C)
- Dodawać nowe konta do zbioru (A)
- Badać poszczególne wpisy do konta (E)

- Modyfikować poszczególne wpisy do konta (M)
- Listować wszystkie konta w zbiorze (L)
- Usuwać konto ze zbioru (D)
- Sortować zbiór (S)

Po aktywowaniu, program ACNT wypisuje opcje i czeka na wybranie jednej z nich. Użytkownik wprowadza wówczas pierwszą literę komendy, która identyfikuje pożądaną opcję (C dla Create-twórca, A dla Add-dodaj, itd).

Program wymaga wprowadzenia dalszych danych lub wyprowadza komunikat zgodnie z wybraną opcją.

Każde konto zawiera następujące dane:

- Kod UIC, który służy jako numer konta
- Hasło
- Domniemane urządzenie systemowe /gdy użytkownik skutecznie przyłączył się do systemu, system przypisuje temu urządzeniu logiczną nazwę SY: /
- Imię
- Nazwisko
- Datę i czas ostatniego przyłączenia się do systemu
- Liczbę przyłączeń użytkownika do systemu.

Użytkownik uprzywilejowany może korzystać z programu ACNT dla zmiany swego hasła.

Dopuszczalnymi znakami w nazwie są: A-Z, 0-9, "x", "u", " " i "!". Znaki dolnego rejestru są zmieniane na odpowiednie znaki górnego rejestru. Każdy znak wierszy od 40 (oktalnie) jest dopuszczalny dla hasła.

3.3.1. Tworzenie katalogu użytkowników (opcja C)

System wieloużytkownikowy nie może mieć równocześnie więcej niż jednego zbioru użytkowników systemu. Bezpośrednio po wygenerowaniu systemu należy utworzyć ten zbiór, aby umożliwić normalne korzystanie z systemu. Spod UIC [1,2] należy uruchomić program ACNT po to, by nieuprzywilejowani użytkownicy nie mieli dostępu do zbioru użytkowników. Program ACNT tworzy zbiór o nazwie [0,0] DOCACC.SYS i określa jego ochronę na

[RWED,RWED,,]. Ochrona ta daje dostęp do zbioru tylko użytkownikom uprzywilejowanym i zadaniom uprzywilejowanym.

Poniższy przykład przedstawia, jak utworzyć zbiór katalogowy:

```
>RUN ACNT<CR>
```

```
* ACCOUNT FILE MAINTENANCE PROGRAM *
```

```
/Program aktualizacji katalogu użytkowników systemu/
```

```
OPTIONS ARE: A-ADD, C-CREATE FILE, D-DELETE,  
E-EXAMINE, L-LIST, M-MODIFY,  
S-SORT, CTRL/Z-EXIT
```

```
/Opcjami są: A-dodawanie, C-tworzenie zbioru,  
D-usuwanie, E-badanie, L-listowanie, M-modyfi-  
kowanie, S-sortowanie, CTRL/Z-zakończenie/
```

```
ENTER OPTION: C<CR>
```

```
/Podaj opcje:/
```

```
ENTER MAXIMUM NUMBER OF ACCOUNTS: n. <CR>
```

```
/Podaj maksymalną liczbę kont/
```

Celem wyspecyfikowania maksymalnej liczby kont, które mogą być umieszczone w zbiorze, użytkownik podaje liczbę dziesiętną (n.). System utworzy wówczas zbiór katalogowy, w którym można pomieścić n kont użytkowników systemu. Jeśli zbiór katalogowy okaże się zbyt mały, to dodatkowy obszar może być dołączony po skopiowaniu zbioru używając PIP z kluczem /BL /należy pamiętać o zachowaniu odpowiedniej protekcji zbioru/.

3.3.2. Rozszerzanie katalogu (opcja A)

Po podaniu przez użytkownika maksymalnej liczby kont, program automatycznie przechodzi do opcji A (dodaj konta) i wypisuje:

```
ENTER ACCOUNT OR <ESC> FOR OPTIONS (N,N):
```

```
/Podaj konto lub ESC dla wprowadzenia opcji
```

```
N,N :
```

W ten sposób program ACNT pozwala użytkownikowi rozpocząć dodawanie kont do zbioru bezpośrednio po jego utworzeniu. W odpowiedzi na to zlecenie, użytkownik podaje albo oktalne numery grupowe i użytkowników dla UIC, oddzielone przecinkiem, (g,m), albo wciska klawisz ESC (przełączenie trybu), aby zażądać powrotu do wyboru opcji. Każde nowe konto jest wprowadzone i zbiór jest automatycznie sortowany według UIC.

Poniższy przykład uruchamia program ACNT, a następnie pokazuje, jak dodawać nowe konta do zbioru.

```
>RUN *ACNT<CR>
ACCOUNT FILE MAINTENANCE PROGRAM
/Program aktualizacji katalogu użytkowników
systemu/

OPTIONS ARE:A-ADD, C-CREATE FILE, D-DELETE,
E-EXAMINE, L-LIST, M-MODIFY, S-SORT, CTRL/Z-
EXIT
/Opcjami są: A-dodawanie, C-tworzenie zbioru,
D-usuwanie, E-badanie, L-listowanie, M-mody-
fikowanie, S-sortowanie, CTRL/Z-zakończenie/

ENTER OPTION: A <CR>
/Podaj opcje: A <CR>

ENTER ACCOUNT OR <ESC> FOR OPTIONS (N,N) :
200,202 <CR>
/Podać konto lub <ESC> dla zmiany opcji (n,n):/

PASSWORD (<=6 CHARS.): ANNA <CR>
/Hasło (<=6 znaków):/

DEFAULT SYSTEM DEVICE (DDU): SY<CR>
/Domniemane urządzenie systemowe (DDU):/

FIRST NAME (<=12 CHARS.): JERZY <CR>
/Imię (<=12 znaków):/

UFD SY00:[200,202]
>
ENTER ACCOUNT OR <ESC> FOR OPTION (N,N)
:200,203 <CR>
```

ACNT wymaga podania kolejnych pozycji danych po wprowadzeniu przez użytkownika numerów grupy i użytkownika UIC. Komunikaty ACNT podają liczbę znaków dozwoloną w haśle, imieniu i nazwisku. Komunikat dotyczący domniemanego urzędnika systemowego wymaga podania nazwy urzędnika i numeru jednostki (opcjonalnie, jeśli numerem jest zero).

Po otrzymaniu nazwiska, ACNT wykonuje komendę UDF MCR celem utworzenia katalogu zbiorów użytkownika na tym urządzeniu. Następnie ACNT powtarza zadanie:

```
ENTER ACCOUNT OR <ESC> FOR OPTIONS (N,N):  
/Podaj konto lub <ESC> w celu zmiany opcji  
n,n :/
```

Użytkownik może kontynuować dodawanie kont, wybrać inną opcję, lub zakończyć program.

3.3.3. Usuwanie kont (opcja D)

Opcja usuwania pozwala użytkownikowi eliminować poszczególne konta ze zbioru. Na przykład: OPTIONS ARE: A-ADD, C-CREATE FILE, D-DELETE, E-EXAMINE, L-LIST, M-MODIFY, S-SORT, CTRL/Z-EXIT

/Opcjami są: A-dodawanie, C-tworzenie zbioru, D-usuwanie, E-badanie, L-listowanie, M-modyfikowanie, S-sortowanie, CTRL/Z-zakończenie/

```
ENTER OPTION: D  
/Podaj opcje:/
```

```
ENTER ACCOUNT OR <ESC> FOR OPTIONS (N,N):  
200,202  
/Podaj konto lub <ESC> w celu zmiany opcji  
(n,n):/
```

```
[200,202] ANNA JERZY NOWAK DBOO:  
12/10/81 20:36:08 00120  
THIS ENTRY ? [Y/N]: N  
/Ten wpis ?/
```

```
[200,202] LIZA NOWAK DBOL:  
12/10/81 23:36:08 00001
```

THIS ENTRY ? [Y/N]: Y

/Ten wpis ?/

DELETE UDF AND FILES ? [Y/N] : Y

/Usunąć UDF i zbiory ?/

PIP DB01:[200,202] *.*;/DE

>PIP DB01:[0,01200202.DIR;*/DE

>

ENTER ACCOUNT OR <ESC> FOR OPTIONS (N,N):

Po wyspecyfikowaniu przez użytkownika konta, które ma być usunięte, program ACNT podaje wartość wpisu i wymaga odpowiedzi Y (tak) lub N (nie) dla weryfikacji. Odpowiedź negatywna powoduje, że program ACNT przeszukuje zbiór katalogowy celem znalezienia innego wpisu o tym samym UIC. Jeśli odpowiedź brzmi Y (tak), program pyta, czy użytkownik chce usunąć UFD konta i zbiory użytkownika. Odpowiedź pozytywna wywołuje komendy dla programu PIP, celem usunięcia tych zbiorów. Jeśli użytkownik podał N w odpowiedzi na pytanie o usuwanie zbiorów, to zbiory i UFD dla tego konta pozostają nie naruszone. Użytkownik może pozostawać wówczas w trybie usuwania podając następne konto lub wprowadzić znak <ESC> dla powrotu do wyboru opcji.

3.3.4. Modyfikowanie kont przez nieuprzywilejowanych użytkowników

Użytkownik nieuprzywilejowany może uruchomić program ACNT celem zmiany swego hasła. Wszystkie pozostałe opcje ACNT są uprzywilejowane. Na przykład:

>RUN ACNT

* ACCOUNT FILE MAINTENANCE PROGRAM *

/Program aktualizacji katalogu użytkowników systemu/

PASSWORD (<=6 CHARS.): DIANA

/Hasło (<=6 znaków.):/

TYPE <ESC> TO LEAVE ENTRY UNCHANGED

/Wprowadź znak <ESC> celem pozostawienia wpisu niezmiennym/

PASSWORD (<=6 CHARS.): EWA

/Hasło (<=6 znaków.):/

OPERATION COMPLETE

/Działanie zakończone/

4. Program wyłączenia systemu operacyjnego

Program SHUTUP jest programem systemowym, który w zorganizowany sposób zatrzymuje system operacyjny DOC PB. Program ten przekazuje do wszystkich przyłączonych do systemu użytkowników odpowiednie komunikaty ostrzegawcze na ich terminale, usuwa wszystkie nieuprzywilejowane zadania i demontuje wszystkie zamontowane urządzenia. Następnie zatrzymuje system. Program SHUTUP może być uruchomiony z dowolnego uprzywilejowanego stanowiska bądź z TTO:.

Następujące zadania muszą być zainstalowane, aby program prawidłowo zatrzymał system:

1. AT., program przetwarzania zbiorów komend MCR
2. ASC, tylko gdy utworzony jest dynamiczny obszar wyładowania
3. BYE, jeśli występuje ochrona wieloużytkownikowa
4. DMO, jeśli są jakieś zamontowane tomy
5. ERF, jeśli aktywne jest zadanie rejestracji błędów

Po uaktywnieniu, program SHUTUP identyfikuje się na terminalu, a następnie prosi o podanie, po ilu minutach system ma być zatrzymany, podanie odstępu czasu, również w minutach, pomiędzy rozgłaszaniem komunikatów o nadchodzącym zatrzymaniu oraz za ile minut ma być zablokowane przyłączenie do systemu.

Na przykład:

>RUN MSHUTUP

DOC PB-2 SHUT DOWN PROGRAM

ENTER MINUTES TO WAIT BEFORE SHUTDOWN: 15
/podaj ile minut oczekiwać przed zatrzymaniem/

ENTER MINUTES BETWEEN MESSAGES: 3
/Podaj ile minut ma upływać pomiędzy komunikatami:/

ENTER MINUTES TO WAIT BEFORE DISABLING

LOGINS: 5
/Podaj ile minut odczekiwać przed zablokowaniem przyłączeń:/

Po otrzymaniu tych danych /liczby 15, 3 i 5/, program SHUTUP prosi o potwierdzenie w następujący sposób:

OK TO SHUTDOWN ? [Y/N]:

/Czy zatrzymać system ?/

Jeśli odpowiedź jest negatywna program SHUTUP kończy działanie. Jeśli pozytywna, rozgłasza następujący komunikat do wszystkich przyłączonych użytkowników na wszystkich terminalach:

PLEASE FINISH UP, 15 MINUTES BEFORE SHUTDOWN

/Proszę skończyć, za 15 minut zatrzymanie/

W 3 minuty później terminale zatrzymują następujący komunikat:

PLEASE FINISH UP, 12 MINUTES BEFORE SHUTDOWN

/Proszę skończyć, za 12 minut zatrzymanie/

Program SHUTUP kontynuuje wydanie komunikatów ostrzegawczych w wyspecyfikowanych odstępach czasu, aż do upłynięcia podanego czasu czekania przed zatrzymaniem.

Jeśli system zawiera ochronę wieloużytkownikową, to po upłynięciu deklarowanego czasu zwłoki przed zablokowaniem przyłączeń /mierzonego od chwili startu SHUTUP/ SHUTUP wyprowadza następującą informację do wszystkich terminali:

ALL FURTHER LOGINS ARE DISABLED

/Zablokowane przyłączenia do systemu/

Kiedy upłynął czas zwłoki przed wyłączeniem systemu SHUTUP wykonuje następujące czynności:

1. Odłącza wszystkich użytkowników /tylko w systemach wieloużytkownikowych/

2. Zatrzymuje drajwer konsoli /jeśli jest w systemie/
3. Przeadresowuje terminal konsoli na TI, który uaktywnił SHUTUP
4. Jeśli jest zbiór LB:[1,2]SHUTUP.COMD to przekazuje go do AT do wykonania
5. Zatrzymuje podsystem wyprowadzania (QUE...).
6. Zatrzymuje zadanie rejestracji błędów
7. Zwalnia wszystkie obszary wyładowania i demontuje wszystkie zamontowane tomy
8. Zawiesza pracę systemu

W tym momencie możemy albo załadować inny system albo wznowić zawieszony system przez wciśnięcie klucza kontynuacji na konsoli CPU. Przy wznowieniu pracy zawieszonego systemu należy zamontować tomy a następnie wykonać systemową procedurę startu.

Przykład:

```
RUN XSHUTUP
```

```
DOC PB-2 SHUT DOWN PROGRAM
```

```
ENTER MINUTES TO WAIT BEFORE SHUTDOWN: 12  
/Podaj ile minut czekać przed zatrzymaniem:/
```

```
ENTER MINUTES BETWEEN MESSAGES:3  
/Podaj ile minut ma upływać pomiędzy komunikatami:/
```

```
ENTER MINUTES TO WAIT BEFORE DISABLING  
LOGINS: 5  
/Podaj ile minut czekać przed zablokowaniem przyłączeń/
```

```
OK TO SHUTDOWN ?[Y/N]: Y  
/Czy wyłączyć system ?/
```

```
PLEASE FINISH UP, 12 MINUTES BEFORE SHUTDOWN  
/Proszę skończyć, za 12 minut zatrzymanie/
```

```
PLEASE FINISH UP, 9 MINUTES BEFORE SHUTDOWN  
/Proszę skończyć, za 9 minut zatrzymanie/
```

```
ALL FURTHER LOGINS ARE DISABLED  
/Zablokowanie przyłączenia do systemu/
```

PLEASE FINISH UP, 6 MINUTES BEFOR SHUTDOWN
/Proszę kończyć, za 6 minut zatrzymanie/

PLEASE FINISH UP, 3 MINUTES BEFORE SHUTDOWN
/Proszę kończyć, za 3 minuty zatrzymanie/

PLEASE FINISH UP, 2 MINUTES BEFORE SHUTDOWN
/Proszę kończyć, za 2 minuty zatrzymanie/

PLEASE FINISH UP, 1 MINUTES BEFORE SHUTDOWN
/Proszę kończyć, za 1 minutę zatrzymanie/

PLEASE FINISH UP, 0 MINUTES BEFORE SHUTDOWN
/Proszę kończyć, za 0 minut zatrzymanie/

@LB:[1,2]SHUTUP

ERF

ERF -- REQUESTED "ERRLOG" TO STOP LOGGING
/Zlecono, by "ERRLOG" zakończył rejestrowanie
błędów urządzeń/

ERL -- LOGGING ENDED AFTER 0 ERRORS
/Rejestrowanie błędów urządzeń zakończone po
0 błędach/

ASC DBO:/BLIKS=0.

ASC -- CHECKPOINT FILE NOW INACTIVE
/Zbiór wyładowania zadań jest teraz nieczynny/

DMO DBO:

***DBO:--DISMOUNT COMPLETE
/Demontaż zakończony/

DMO DB1:

***DB1:--DISMOUNT COMPLETE
/Demontaż zakończony/

ASC DSO:/BLKS=0.

ASC--CHECKPOINT FILE NOW INACTIVE
/Zbiór wyładowania zadań jest teraz nieczynny/

DMO DSO:/DEV

***DSO: -- DISMOUNT COMPLETE
/Demontaż zakończony/

DMO MMO: PAYROL

MTAACF -- MMO: ** DISMOUNT COMPLETE **
/Demontaż zakończony/

Celem zatrzymania systemu w tym przykładzie,
program SHUTUP:

- Wywołuje zbiór komend [1,2]SHUTUP.COMD, która z kolei usuwa zadanie ATTT1
- Uruchamia zadanie ERF celem zakończenia zadania rejestrowania błędów (ERL). /Patrz "Rejestracja błędów sprzętu"/.
- Anuluje przydział zbioru wyładowania zadań na DBO: i DSO: (na przykład, ASC DBO:/BLKS=0)
- Demontuje dyski na DBO:, DB1: oraz DSO: (na przykład, DMO DBO:),
- Demontuje taśmę magnetyczną o etykiecie PAYROL na MMO: (DMO MMO:).

Program SHUTUP wydaje także komendy BYE dla wszystkich przyłączonych do systemu użytkowników na ich terminalach, z wyjątkiem terminala, z którego aktywowany był SHUTUP. Komenda BYE, po odłączeniu terminala wyprowadza na nim komunikat:

>

HAVE A GOOD AFTERNOON

22-JAN-79 3:49 TT2: LOGGED OFF

KOMENDY MONITORA

Wystąpienie litery P oznacza, że format komendy lub słowa kluczowe są uprzywilejowane.

Nieuprzywilejowane komendy monitora

ABO nazwa-zadania [/słowo-kliczowe]

Słowo kluczowe: PMD

Powoduje zakończenie wykonania specyfikowanego zadania. Opcjonalne wyspecyfikowanie PMD generuje obraz pamięci zadania.

AST [/słowo-kluczowe]

Słowo kluczowe: ALL

TERM=TTnn:

Wyprowadza na terminal nazwy zadań uaktywnionych z danego terminala. W wypadku opcji ALL wyprowadza nazwy wszystkich aktywnych zadań w systemie. Przy słowie kluczowym TERM komenda wyprowadza nazwy zadań uaktywnionych z danego terminala.

ALL ddn:[słowo-kluczowe]

Słowo kluczowe: =LLnn:

/TERM=TTnn:

Określa wyspecyfikowane urządzenie jako urządzenie prywatne użytkownika. Uprzywilejowani użytkownicy mogą przywiązywać urządzenie do danego terminala wykorzystując słowo kluczowe TERM, użytkownicy nieuprzywilejowani mogą przywiązywać urządzenie tylko do własnego terminala. Użycie słowa kluczowego =LLnn: zezwala użytkownikowi na utożsamianie urządzenia fizycznego ddnn: z urządzeniem logicznym LLnn:. Specyfikacja urządzenia bez podania numeru jednostki powoduje przypisanie pierwszego wolnego urządzenia danego typu.

ASN ppn:=LLnn:[/słowo-kluczowe]

Słowo kluczowe: GBL

LOGIN

TERM=TTnn:

Definiuje, wyprowadza lub usuwa przypisania urządzeń logicznych.

Przypisania lokalne

ASN ppn:=LLnn:

ASN ppn:=LLnn:/TERM=TTnn:(P)

Przypisania na czas przyłączenia do systemu

ASN ppn:=LLnn:/LOGIN (P)

ASN ppn:=LLnn:/LOGIN/TERM=TTnn:(P)

Przypisania globalne

ASN ppn:=LLnn:/GBL (P)

Wyprowadzenie przypisań lokalnych

ASN

Wyprowadzenie przypisań na czas przyłączenia do systemu

ASN /TERM=TTnn:(P)

Wyprowadzenie przypisań globalnych

ASN /GBL (P)

Usunięcie przypisań lokalnych

ASN =

ASN =LLnn:

Usunięcie przypisań na czas przyłączenia do systemu

ASN =/LOGIN (P)

ASN =/LOGIN/TERM=TTnn:(P)

ASN =/TERM=TTnn:(P)

ASN =/LLnn:/TERM=TTnn:(P)

Usunięcie przypisań globalnych

ASN =/GBL

ASN =LLnn:/GBL (P)

ALT [nazwa-zadania]

Wyprowadza na terminal nazwę i stan wszystkich aktywnych zadań w systemie lub stan specyfikowanego zadania.

BRO TTnn:komunikat
BRO @specyfikacja-zbioru
BRO ALL:komunikat (P)
BRO LOG:komunikat (P)

Wyprowadza specyfikowany komunikat na terminalu TTnn: lub dla użytkowników uprzywilejowanych na wszystkich lub przyłączonych terminalach.

BYE

Informuje system o zakończeniu pracy przez użytkownika.

CAN nazwa-zadania

Anulowanie zależnego od czasu zlecenia wykonania zadania (p. komenda RUN). Nieuprzywilejowani użytkownicy mogą anulować tylko te zlecenia, które sami wydali.

DEA [ddn:]

Unieważnia przydział urządzenia ddn: jako urządzenia prywatnego. O ile nie wyspecyfikowano urządzenia komenda dotyczy wszystkich urządzeń prywatnych użytkownika.

DEV [/słowo-kluczowe]

Słowo kliczowe: dd:
LOG

Wyprowadza symboliczne nazwy wszystkich urządzeń lub nazwy wszystkich urządzeń wskazanego typu dd:. Słowo kluczowe LOG powoduje określenie, które z terminali są przyłączone do systemu.

DMO ddn:[etykieta][/słowa-kluczowe]

Słowa kluczowe: DEV (P)
USER
TERM=TTnn:(P)

FLA [ggg/słowo-kluczowe]

Słowa kluczowe: CRE
ELIM

Uprzywilejowani użytkownicy mogą tworzyć lub eliminować globalne wskaźniki dla każdej grupy użytkowników. Użytkownicy nieuprzywilejowani mogą to czynić tylko dla własnej grupy.

HEL UIC/HASŁO

Przyłączenie użytkownika do systemu wieloużytkownikowego.

HELP [kwalifikator1][kwalifikator2]...[kwalifikator9]

HELP %[kwalifikator1][kwalifikator2]...[kwalifikator9]

Wyprowadza na terminal zawartość zbioru [1,2] HELP.HLP lub zbioru HELP.HLP znajdującego się w katalogu użytkownika.

INI ddn:etykieta-tomu [/słowa-kluczowe]

Słowa kluczowe:

BAD=[opcja] sposób wprowadzania bloków uszkodzonych
=[AUTO] nych
=[MAN]
=[AUTO,MAN]
=[OVR]
=[OVR,MAN]

DENS=gęstość-zapisu

EXT=ilość-bloków-rozszerzenia-zbioru

FPRO=[system,właściciel,grupa,ogól] prawa dostępu do zbiorów tomu

INDX=pozycja-zbioru-nagłówków
=BEG
=MID

=END
=BLK:nnn
INF=początkowa-wielkość-zbioru-nagłówków
LRU=ilość-zbiorów-tomu
PRO=[system,właściciel,grupa,ogól] prawa dostę-
pu do tomu (nie zaimplementowane)
UIC=[grupa,użytkownik]
WIN=ilość-wskaźników-odszukania-dla-okien-zbio-
ru
VI= wyprowadza listę wszystkich słów kluczowych
Tworzy tom o strukturze zbiorów DOC PB na dysku
lub tomie.
LUN nazwa-zadania
Wyprowadza na terminal listę LUN dla specyfiko-
wanego zadania.
MOU ddn:[etykieta-tomu][słowa-kluczowe]
Słowo kluczowe:
ACP=nazwa-zadania-ACP
DENS=gęstość-zapisu
EXT=ilość-bloków-rozszerzenia-zbioru
FOR= tom obcy
FPRO=[system,właściciel,grupa,ogól]
LRU=ilość-bloków-FCB
OVR ominięcie sprawdzenia etykiety
PARM="parametry-użytkownika"
UIC= kod-identyfikacyjny-użytkownika
UNL odblokowanie zbioru nagłówków
VI informacje o tomie
WIN=ilość-wskaźników-odszukania-dla-okien-zbio-
ru
Tworzy blok VCB i umożliwia dostęp do tomu
przez system zbiorów
Format dla taśmy magnetycznej:
MOU ddn:etykieta[słowa-kluczowe]
Słowa kluczowe:
ACP=nazwa-zadania-ACP
DENS=gęstość-zapisu

FOR= tom obcy
FPRO=[system,właściciel,grupy,ogól]
OVR ominięcie sprawdzania etykiety
OVRFSID
OVREXP
PARAM="parametry-użytkownika"
UIC=[kod-użytkownika]
VI wprowadzenie informacji o tomie

PAR

Wyprowadza na terminal informacje o każdej partycji w systemie.

RES nazwa-zadania

Zezwala nieuprzywilejowanym użytkownikom kontynuować wykonanie zawieszonoego zadania, które było zainicjowane z danego terminala. Użytkownicy uprzywilejowani mogą kierować komendę RES do każdego zawieszonoego zadania.

RUN nazwa-zadania [/RSI=przerwa] [/UIC=[uic]]

RUN nazwa-zadania opóźnienie [/RSI=przerwa]
[/UIC=[uic]]

RUN nazwa-zadania synch [opóźnienie] [/RSI=przerwa]
[/UIC=[uic]]

RUN nazwa-zadania czas-absolutny [/RSI=przerwa]
[/UIC=[uic]]

RUN [ddn:] [x]specyfikacja-zbioru [/słowa-kluczowe]

Słowa kluczowe:

CKP=opcja opcja przeładowania
=YES
=NO

INC=wielkość

PAR=nazwa-partycji

PRI=liczba priorytet zadania

PMD=opcja wyprowadzanie obrazu pamięci PMD
=YES
=NO

SLV=opcja opcja podporządkowania zadania
=YES
=NO

TASK=nazwa-zadania

UIC=[g,u]

Rozpoczyna wykonanie wyspecyfikowanego zadania, natychmiast lub w określonym czasie.

przerwa - przerwa między wywołaniami zadania przy opcji wielokrotnego uruchamiania

opóźnienie - czas po jakim od chwili wydania komendy zostanie zainicjowane zadanie

synch - jednostka czasu synchronizująca zainicjowanie wykonania zadania

SET /słowa=kluczowe=wartość

Słowa kluczowe:

BUF=ddn:[rozmiar] wielkość bufora specyfikowanego urządzenia

CRT[=TTnn:] terminal TTnn: jest terminalem ekranowym

EBC[=TTnn:] do bufora terminala TTnn: jest przekazywane pełne 8 bitów znaku

ECHO[=TTnn:] wyprowadzanie każdego znaku wprowadzonego z terminala

ESCSEQ[=TTnn:] rozpoznawanie sekwencji ESC przez specyfikowany terminal

FDX[=TTnn:] określa drajwer terminala TTnn: jako duplexowy

FORMFEED[=TTnn:] terminal TTnn: posiada sprzętową realizację wysuwu strony

HFILL=TTnn:[wartość] określa ilość znaków dopełniających dodanych po wprowadzeniu znaku powrotu karetki <CR> na terminal TTnn:

HHT[=TTnn:] terminal TTnn: posiada sprzętową realizację tabulacji poziomej

HOLD[=TTnn:]	dla terminala ekranowego TTnn: odblokowuje tryb stronicowania wyświetla- nia
LINES=TTnn:[wartość]	określa ilość wierszy na stronie dla terminala TTnn:
LOGON (P)	umożliwia przyłączenie się do systemu przez użytkownika
LOWER[=ddn:]	znaki dolnego rejestru nie będą przekształcane na znaki rejestru górne- go
MAIN=nazwa[:baza:rozmiar:typ](P)	ustawienie partycji głównej
MAXEXT[=rozmiar] (P)	określa maksymalny roz- miar, do jakiego zadanie może się samo rozszerzyć
MAXPKT[=n] (P)	określa maksymalną ilość pakietów przydzielonych dla optymalizacji WE/WY
POOL[=adres] (P)	określa adres w pamięci, do jakiego mogą się za- czynać partycje użytkow- nika
PRIV[=TTnn:]	ustawia specyfikowany terminal w stan uprzywi- lejewania
PUB[=ddn:]	określa specyfikowane u- rządzenie jako urządzenie ogólnodostępne
REMOTE[=TTnn:]	określa specyfikowany ter- minal jako terminal od- legły
RPAC[=TTnn:]	drajwer terminala TTnn: będzie czytał do bufora wszystkie znaki, łącznie ze znakami funkcji spec- jalnych

SLAVE[=TTnn:] określa terminal TTnn:
jako podporządkowany

SPEED=TTnn:[odbiór:nadawanie] ustawia dla
terminala TTnn: szybkość
transmisji nadawania i
odbioru

SUB=nazwa-partycji:nazwa-podpartycji[:baza:roz-
miar] (P) ustawienie podpartycji

SYSUIC[=uic] (P) ustawia systemowy UIC

TERM=TTnn:[typ-terminala] ustawia typ termi-
nala TTnn:

TYPEAHEAD[=TTnn:] dane czytanie z terminala
TTnn: będą buforowane
przed przesłaniem do za-
dania

UIC[=uic[:TTnn:]] ustawia domniemany UIC
dla terminala TTnn: lub
terminala wydającego ko-
mendę

VFILL[=TTnn:] dołączenie 4 znaków pus-
tych po każdym znaku no-
wej linii <LF>

WCHKC =ddn:] każda operacja zapisu na
specyfikowane

urządzenie
będzie przeprowadzana ze
sprawdzeniem jej popraw-
ności

WRAP[=TTnn:]
TTnn:
w przypadku przepełnienia
bufora terminala

automatyczne generowanie
przejścia do

nowego
wiersza

Komenda powoduje określenie lub zmianę cech u-
rządzeń. Uprzywilejowani użytkownicy mogą zmie-
nić cechy wszystkich urządzeń w systemie. Użytkownicy nieuprzywilejowani mogą tylko zmienić
cechy urządzeń im przydzielonych.

STP [nazwa-zadania]

Wyprowadza nazwę i status wszystkich zadań zainstalowanych w systemie lub zadania wyspecyfikowanego.

TAS

Opisuje każde zadanie zainstalowane w systemie.

TIM [dd-mmm-rr][gg:mm[:ss]] lub [mm/dd/rr]
[gg:mm[:ss]]

Wyprowadza aktualny czas i datę. Użytkownicy uprzywilejowani mogą również zmienić te wartości.

UDF ddn:[etykieta-tomu][g,u][słowa-kluczowe]

Słowa kluczowe:

ALLOC=liczba ilość wpisów do katalogu

PRO=[s,w,g,e] prawa dostępu do katalogu

Tworzy katalog UFD na danym tomie i wprowadza go do głównego katalogu MFD. Uprzywilejowani użytkownicy mogą tworzyć UFD na każdym tomie, użytkownicy nieuprzywilejowani tylko na swoich tomach prywatnych.

UNS [nazwa-zadania]

Wznawia wykonanie wcześniej zatrzymanego zadania.

Uprzywilejowane komendy monitora

ASC ddn:/BLKS=n

Przydziela obszar wylądowania na dysku. Jest to zbiór o wielkości nbloków.

ALT nazwa-zadania[/słowa-kluczowe]

Słowa kluczowe:

PRI= priorytet-bieżący-i-statyczny

RPRI= tylko-priorytet-bieżący

Zmienia statyczny lub bieżący priorytet zainstalowanego zadania.

BOO [specyfikacja-zbioru]

Ładuje system istniejący jako zbiór obrazu zadania w tomie.

BRK

Przekazuje sterowanie do programu oczyszczania egzekutora XDT.

CLQ

Wyprowadza informacje o zadaniach aktualnie znajdujących się w kolejce zegarowej. Są to zadania uruchamiane komendą RUN lub dyrektywą RUN z wyspecyfikowanymi opcjami uzależniającymi uruchomienie zadania od czasu.

FIX nazwa-zadania

Ładuje i umieszcza zadanie w pamięci na stałe.

INS [x]specyfikacja-zbioru[/słowa-kluczowe]

Słowa kluczowe:

CKP=opcja opcja wyładowania

 =YES

 =NO

EST=opcja opcja stanu wyjścia

 =YES

 =NO

INC=rozmiar dodatkowy obszar pamięci przydzielony zadaniu

LCA=opcja konwersja znaków dolnego rejestru na rejestr górny

 =NO

PAR=nazwa-partycji

PMD=opcja wprowadzanie obrazu pamięci post-mortem

 =YES

 =NO

PRI=liczba priorytet zadania

SLV=opcja podporządkowanie zadania

 =YES

 =NO

TASK=nazwa-zadania

UIC=[g,a] domniemany UIC zadania

Instaluje dane zadanie w systemie.

LOA ddn:[/słowa-kluczowe]

Słowa kluczowe: PAR=nazwa-partycji
SIZE=wielkość-partycji
HIGH

Ładuje nierezidentny drajwer urządzenia do pamięci.

OPE adres-pamięci [+/-n][/słowo-kluczowe]
(adres-pamięci)(zawartość-pamięci)
[/wartość]<ogranicznik-wiersza>

Słowa kluczowe:

TASK=nazwa-zadania dostęp do zadania
PAR=nazwa-partycji dostęp do partycji
KNL dostęp do pierwszych 32K
 pamięci
DRV=dd: dostęp do ładowalnego
 drajwera

Pozwala sprawdzić i modyfikować słowo pamięci.

REA nazwa-zadania lun ddn:

Zmiana przypisania logicznego numeru urządzenia w zadaniu z jednego fizycznego urządzenia na inne.

RED nddn:=oddn:

Przeadresowuje wszystkie zlecenia I/O z jednej fizycznej jednostki urządzenia na inną (z 'o' na 'n').

REM nazwa-zadania

Usuwa zadanie (jego nazwę) z STD i w efekcie wycofuje dane zadanie z systemu.

SAV [/słowa-kluczowe]

Słowa kluczowe: WB

MOU="tekst"

SFILE=specyfikacja-zbioru

Kopiuje bieżący obraz systemu do zbioru obrazu systemu, z którego nastąpiło załadowanie systemu.

SSM tekst-komunikatu

Wprowadza tekst komunikatu do raportu programu rejestracji błędów.

UNF nazwa-zadania

Usuwa zadanie z pamięci, gdzie było umieszczone po komendzie **FIX**.

UNL dd;

Usuwa ładowalny drajwer urządzenia z pamięci.

PROGRAM PRZETWARZANIA ZBIORÓW KOMEND

Wyróżniamy dwa typy zbiorów komend: zbiór komend zadania i zbiór komend MCR.

Zbiór komend zadania zawiera listę komend określonego zadania. Raz wprowadzona i umieszczona w zbiorze sekwencja komend może być wielokrotnie wywoływana poprzez wypisanie nazwy zbioru. W celu zainicjowania zbioru komend zadania, należy w wierszu komendy po nazwie zadania umieścić nazwę zbioru komend poprzedzoną znakiem @. Zadanie zlecające wprowadzenie informacji uzyskuje wówczas dostęp do wyspecyfikowanego zbioru i rozpoczyna czytanie go i reaguje na komendy w nim zawarte. Na przykład, celem zainicjowania zbioru komend makroasemblera, użytkownik wprowadza:

```
>MAC @INPT.CMD
```

Makroassembler uzyskuje wówczas dostęp do zbioru INPT.CMD i wykonuje komendy w nim zawarte. Dominującym typem zbioru dla zbiorów komend jest .CMD. Zbiór komend MCR zawiera listę komend MCR. Może on także zawierać komendy specjalne pozwalające na sterowanie wykonywaniem zbioru komend.

Celem zainicjowania zbioru komend MCR, użytkownik po prostu wprowadza nazwę zbioru, poprzedzoną znakiem(@). Na przykład:

```
>@MCRIP
```

Niektóre zadania, w tym także MCR, dopuszczają stosowanie zagnieżdżonych zbiorów komend (jeden zbiór komend odwołuje się do drugiego). Maksymalna dopuszczalna głębokość zagnieżdżenia zależy od zadania i podana jest w dokumentacji poszczególnych zadań.

Gdy procesor osiąga koniec zbioru najwyższego poziomu, wprowadza komunikat:

```
>@<EOF>
```

po czym kończy przetwarzanie zbioru.

W celu wcześniejszego wyjścia z procesora zbiorów komend należy w odpowiedzi na zlecenie wprowadzenia informacji wcisnąć CTRL/Z.

1. Procesor zbiorów komend MCR

Większość zadań odczytuje i reaguje na komendy zawarte w zbiorze komend w taki sposób, jakby komendy te zostały wprowadzone bezpośrednio z terminala.

Zbiory komend MCR mogą wywoływać inne zbiory komend MCR, z tym, że maksymalna głębokość zagnieźdżenia jest ograniczona do czterech poziomów zbiorów. Przy przetwarzaniu zbioru komend, MCR wyprowadza na terminal, z którego zainicjowano przetwarzanie, każdą wykonaną komendę MCR oraz wszystkie komentarze, które pojawiają się w zbiorze.

W dyrektywach procesora pierwszym znakiem w wierszu jest kropka(.). Komendy MCR nie mają na początku specjalnego znaku.

Dyrektywy AT. tworzą język proceduralny, który pozwala na:

1. Definiowanie etykiet
2. Definiowanie i nadawanie wartości symbolom trzech typów: logiczne, liczbowe i tekstowe
3. Tworzenie i dostęp do zbiorów danych
4. Sterowanie wykonywaniem dyrektyw zbiorów komend
5. Przeprowadzanie testów logicznych
6. Blokowanie i odblokowywanie każdego z sześciu trybów pracy
7. Zwiększanie i zmniejszanie wartości symboli liczbowych
8. Sterowanie równoległym i uwarunkowanym czasowo wykonaniem zadań

Komentarze, które rozpoczynają wiersz interpretowany przez MCR, poprzedzone są średnikiem (;). Komentarze, które pojawiają się w wierszu za komendą MCR, są poprzedzane znakiem wykrzyknika (!).

Poniżej podany jest opis wszystkich dyrektyw AT. z podziałem na funkcje jakie wykonują. Dokładny opis wybranych dyrektyw znajduje się w następnym punkcie.

Definiowanie etykiety

- .etykieta:** - Przypisz nazwę do wiersza w zbiorze komend tak, żeby można było się do niego odwołać.

Definiowanie symboli

- .ASK** - Zdefiniuj lub przedefiniuj symbol logiczny i przypisz mu wartość logiczną (prawda lub fałsz).
- .ASKN** - Zdefiniuj lub przedefiniuj symbol liczbowy i przypisz mu wartość liczbową.
- .ASKS** - Zdefiniuj lub przedefiniuj symbol tekstowy i przypisz mu wartość ASCII.
- .ERASE** - Usuń definicje symbolu lokalnego lub globalnego.
- .SETT/.SETF** - Zdefiniuj lub przedefiniuj symbol i nadaj mu wartość logiczną (prawda lub fałsz).
- .SETN** - Zdefiniuj lub przedefiniuj symbol i przypisz mu wartość liczbową.
- .SETS** - Zdefiniuj lub przedefiniuj symbol i przypisz mu wartość tekstową.

Dostęp do zbiorów

- .DATA** - Określa wiersz danych, który należy wprowadzić do wyjściowego zbioru danych.
- .CHAIN** - Zamyka aktualny zbiór komend i pobiera komendę z innego zbioru.
- .CLOSE** - Zamyka wyjściowy zbiór danych.
- .OPEN** - Tworzy i otwiera wyjściowy zbiór danych (jeśli zbiór istnieje, tworzy nową wersję i otwiera ją).
- .OPENA** - Otwiera istniejący zbiór i dopisuje sekwencje danych (nie tworzy nowej wersji).

Sterowanie logiczne

- .BEGIN - Oznacza początek bloku.
- .END - Oznacza koniec bloku.
- .EXIT - Kończy przetwarzanie bloku lub aktualnego zbioru komend, przekazując sterowanie do wyższego poziomu i opcjonalnie ustawia wartość symbolu specjalnego <EXSTAT>.
- .GOSUB - Skok do etykiety określającej podprogram wewnątrz zbioru komend.
- .GOTO - Skok do etykiety wewnątrz zbioru komend.
- .ONERR - Skok do etykiety w przypadku wykrycia błędu.
- .RETURN - Wyjście z podprogramu i powrót do wiersza następnego po wywołaniu podprogramu.
- .STOP - Kończy przetwarzanie zbioru komend opcjonalnie ustawia stan zakończenia AT.

Testy logiczne

- .IF - Sprawdza, czy symbol spełnia jeden lub szereg możliwych warunków.
- .IFACT/.IFNACT - Sprawdza, czy zadanie jest aktywne.
- .IFDF/.IFNDF - Sprawdza, czy symbol jest zdefiniowany.
- .IFINS/.IFNINS - Sprawdza, czy zadanie jest zainstalowane.
- .IFLOA/.IFNLOA - Sprawdza, czy drajwer urządzenia jest załadowany.
- .IFT/.IFF - Sprawdza, czy symbol logiczny ma wartość prawda, czy fałsz.
- .TEST - Testuje długość ciągu tekstowego.

- Blokowanie i odblokowywanie trybów wykonania
- .DISABLE - Zablokowanie opcji podstawiania, wy-
prowadzania danych, definiowania
symboli globalnych, dolnego rejest-
ru, rozpoznawania znaku <ESC> lub
kontrolera echa.
 - .ENABLE - Odblokowanie opcji podstawiania, wy-
prowadzania danych, definiowania
symboli globalnych, dolnego rejest-
ru, rozpoznawania znaku <ESC> lub
kontrola echa.

Zwiększenie i zmniejszenie wartości liczbowych
symboli

- .DEC - Zmniejsz wartość liczbową o jeden.
- .INC - Zwiększ wartość liczbową symbolu o
jeden.

Sterowanie wykonaniem

- .DELAY - Opóźnia wykonanie zbioru komend o
określony wymiar czasu.
- .PAUSE - Zatrzymuje procesor zbioru komend i
dopuszcza akcje użytkownika.
- .WAIT - Czeka na zakończenie określonego za-
dania.
- .XQT - Uruchamia zadanie, przysyła mu
wiersz komendy i kontynuuje przetwa-
rzanie zbioru nie czekając na zakoń-
czenie zadania.

2. Symbole

Procesor zbiorów komend MCR pozwala użytkowni-
kowi definiować symbole. Nazwy symboli są cią-
gami znaków ASCII o długości jednego do sześć -
ciu znaków. Muszą one zaczynać się od litery
lub znaku dolara(\$). Pozostałe znaki muszą być
alfanumeryczne lub *.

Są trzy typy symboli:

- Logiczny
- Liczbowy
- Tekstowy

Symbol logiczny może przyjąć wartość prawda lub fałsz.

Wartością symbolu tekstowego jest ciąg znaków ASCII o długości od 0 do 80/10/ znaków.

Symbol liczbowy może przyjmować wartości z przedziału 0 - 65535/10/. Symbol można zdefiniować jako liczbę dziesiętną albo jako oktalną. Wartość /dziesiętna lub oktalna/ jest istotna tylko, gdy symbol jest podstawiany.

Typ symbolu /logiczny, liczbowy, tekstowy/ jest zdefiniowany przez pierwszą dyrektywę przypisującą wartość symbolowi. Następne przypisania mogą zmienić wartość, ale nie mogą zmienić typu symbolu. Dyrektywy przypisujące mogą przypisać:

- Wartość "prawdy" bądź "fałszu", przy definiowaniu symbolu logicznego. (Definiowane przez .ASK, .SETT lub .SETF.)
- Wartość oktalna lub dziesiętna, przy definiowaniu symbolu liczbowego. (Definiowane przez .ASKN, SETN.)
- Ciąg znaków dla zdefiniowania symbolu tekstowego. (Definiowane przez .ASKS lub .SETS.)

2.1. Symbole specjalne

Procesor zbioru komend automatycznie definiuje symbole specjalne zależne od określonych własności systemu i odpowiedzi na pytania postawione podczas przetwarzania zbioru komend. Wszystkie symbole specjalne mają wspólny format, nawiasy kątowe(<>) zawierają nazwę symbolu specjalnego.

2.1.1. Logiczne symbole specjalne

Logiczne symbole specjalne mają przypisane wartości prawda lub fałsz w następujących warunkach.

<ESCAPE> Przyjmuje wartość prawda, jeśli na ostatnie pytanie odpowiedzią był znak <ESC>. W przeciwnym przypadku przyjmuje wartość fałsz.

- <DEFAULT> Przyjmuje wartość prawda, jeśli na ostatnie pytanie (.ASKN) odpowiedź była domniemana (wprowadzony był pojedynczy znak <CR>).
- <ALPHAN> Przyjmuje wartość prawda, gdy ostatni tekst wprowadzony w odpowiedzi na dyrektywę .ASKS lub testowany dyrektywą .TEST zawierał tylko znaki alfanumeryczne. Odpowiedź na pytanie wprowadzona małymi literami, kiedy dolny rejestr jest zablokowany nadaje symbolowi <ALPHAN> wartość fałsz. Tekst pusty nadaje symbolowi <ALPHAN> wartość prawda.
- <RAD50> Przyjmuje wartość prawda, gdy ostatni tekst wprowadzony w odpowiedzi na dyrektywę .ASKS lub testowany dyrektywą .TEST zawierał tylko znaki Radix-50. Znaki Radix-50 są to znaki alfanumeryczne plus kropka i znak dolara. Spacja nie jest znakiem Radix-50 w tym kontekście. Tekst pusty także nadaje wartość prawda symbolowi <RAD50>.
- <MAPPED> Przyjmuje wartość prawda, jeśli system, w którym AT. pracuje jest mapowany, fałsz, jeśli system jest nie-mapowany.

2.1.2. Liczbowe symbole specjalne

Liczbowe symbole specjalne przyjmują następujące wartości.

- <MEMSIZ> Przyjmuje wartość rozmiaru pamięci aktualnego systemu w K słowach.
- <SYUNIT> Przyjmuje wartość numeru jednostki domniemanego urządzenia użytkownika (SY:)
- <STRLEN> Przyjmuje wartość długości tekstu wprowadzonego na zlecenie ostatniej dyrektywy .ASKS lub tekstu testowanego przez ostatnią dyrektywę .TEST.

2.1.3. Tekstowe symbole specjalne

Tekstowe symbole specjalne przyjmują następujące wartości.

- <DATE> Aktualna data w formacie dd-mmm-rr (dd-dzień, mmm-miesiąc, rr-rok).
- <SYDISK> Dwuliterowa nazwa urządzenia, które jest domniemanym urządzeniem użytkownika (SY:) w formacie DK:.
- <SYUIC> UIC systemowe w formacie [ggg,mmm].
- <TIME> Aktualny czas w formacie g:min:sec.
- <UIC> Aktualny UIC w formacie [ggg,mmm].

2.2. Podstawienie wartości symbolu

Komendy MCR mogą stosować wartości przypisane do logicznego, liczbowego, tekstowego lub specjalnego symbolu przez zastępowanie parametru /na przykład jednostki urządzenia/, nazwą symbolu zamkniętą w apostrofy /na przykład, 'DEV'/.

Gdy poprzednio dyrektywa odblokowała tryb podstawiania (.ENABLE SUBSTITUTION), to AT, zastępuje nazwę symbolu zamkniętego w apostrofy ciągiem znaków przypisanych symbolowi, przeszukuje wówczas tablice symboli w celu znalezienia wartości odpowiadającej danemu symbolowi i podstawia tę wartość do wiersza komendy na miejsce nazwy symbolu i ograniczających apostrofów.

Na przykład, w zbiorze komend pojawiają się niżej pokazane trzy wiersze. Gdy procesor przetwarza te wiersze, wyprowadza na terminal dwa ostatnie wiersze przykładu.

```
.ASKS DEV PODAJ NAZWĘ URZĄDZENIA  
.ENABLE SUBSTITUTION  
MOUNT 'DEV'
```

```
>* PODAJ NAZWĘ URZĄDZENIA [S]: DK2:  
>MOUNT DK2:
```

DK2: było odpowiedzią na wprowadzone pytanie. Odpowiedź ta przypisała wartość tekstową DK2: symbolowi nazwanemu DEV. Później, gdy procesor zbiorów komend odczyta:

```
MOUNT 'DEV'
```

'to nastąpi 'DEV' wartością przypisania do DEV, czyli, DK2: . Jeśli tryb podstawiania nie był odblokowany, to operator przekazałby ten wiersz do MCR ta, jak pojawił się on w zbiorze komend (to jest, MOUNT 'DEV').

2.3. Symbole i wyrażenia liczbowe

Symbol liczbowy jest ciągiem cyfr odpowiadających wartości z przedziału 0-177777 oktalnie (0-65535 dziesiętnie, jeśli zakończony jest kropką). Symbole liczbowe lub stałe mogą być łączone z innymi symbolami liczbowymi lub stałymi poprzez operatory logiczne lub arytmetyczne tworząc wyrażenia liczbowe. W wyrażeniach mogą występować następujące operatory: dodawanie (+), odejmowanie (-), mnożenie (\times) i dzielenie (/). Operatory logiczne to: iloczyn logiczny (!) i suma logiczna(&). Niedozwolone są spacje i tabulacje pomiędzy operatorami. Operacje arytmetyczne w wyrażeniach liczbowych wykonywane są od lewej do prawej (bez priorytetów), tylko podwyrażenia ujęte w nawiasy wykonywane są jako pierwsze. Na przykład wiersze komendy:

```
.SETN N1 2  
.SETN N2 3  
.SETN N3 N1+N2*4
```

przypisuje symbolowi liczbowemu N3 wartość oktalną 24, podczas gdy komendy

```
.SETN N1 2  
.SETN N2 3  
.SETN N3 N1+(N2*4)
```

przypisują symbolowi liczbowemu N3 wartość oktalną 16.

Typ wyrażenia jest oktalny, jeśli wszystkie operandy są oktalne. W przeciwnym przypadku typ jest dziesiętny.

Na przykład wiersze komendy:

```
.SETN N1 2  
.SETN N2 3  
.SETN N3 N1+N2*4.
```

przypisują symbolowi liczbowemu N3 wartość dziesiętną 20.

Wyrażenia liczbowe mogą występować jako drugi argument w dyrektywach .IF i .SETN. Są one także dopuszczalne jako granice i argumenty domniemane w dyrektywach .ASKN i .ASKS. Dyrektywy .EXIT i .STOP pozwalają na użycie wyrażień liczbowych określających stan zakończenia zadania.

2.3.1. Symbole tekstowe, fragmenty tekstu i wyrażenia

Stała tekstowa jest ciągiem dowolnych znaków ograniczonych cudzysłowami. Dopuszczalne są teksty puste. Ilość znaków nie może przekraczać 80.

Przykład:

```
"ABCDEF"
```

Symbole tekstowe mogą przyjmować wartość dowolnej stałej tekstowej. Wartość można przypisać używając dyrektywy .SETS lub .ASKS. Na przykład wiersze komendy:

```
.SETS S1 "ABCDEF"
```

```
.SETS S2 S1
```

przypisują symbolowi tekstowemu S2 wartość symbolu tekstowego S1 (tzn. ABCDEF).

Możliwe jest dzielenie wartości symbolu tekstowego na fragmenty. Fragmentu można użyć jako drugiego operandu dyrektywy .SETS i .IF. Na przykład wiersze komendy:

```
.SETS S1 "ABCDEF"
```

```
.SETS S2 S1[1:3]
```

przypisują symbolowi tekstowemu wartość znaków od pierwszego do trzeciego symbolu S1 (to jest ABC).

Można tworzyć wyrażenia tekstowe jako kombinacje stałych, symboli i podtekstów z innymi stałymi tekstowymi, symbolami i podtekstami połączonymi operatorem łączenia tekstów (+).

Na przykład wiersze dyrektyw:

```
.SETS S1 "A"
```

.SEZS S2 "CDE"

.SETS S3 S1+"B"+S2[1:3]

przypisują symbolowi tekstowemu S3 wartość połączoną symbolu tekstowego S1, stałej tekstowej "B" i pierwszych trzech znaków symbolu tekstowego S2 /to znaczy ABCDE/.

3. Klucze

Procesor zbiorów komend akceptuje trzy klucze: /TR, /DE i /MC.

/TR - Wyprowadza na terminal ślad każdego przetwarzanego wiersza. Funkcja ta jest przydatna dla oczyszczania zbioru komend. Jeśli komenda powoduje podjęcie jakiegoś działania, to następny drukowany wiersz opisze to działanie. Zazwyczaj taki wiersz składa się z komend MCR wydanych w wyniku poprzedniej dyrektywy.

/DE - Oznacza, że po zakończeniu przetwarzania należy usunąć ten zbiór komend.

/MC - Oznacza, że komendy są przesyłane do MCR. Klucze określone w komendzie inicjującej przetwarzanie zbioru komend są domniemaniami podczas przetwarzania komend @filename i dyrektyw .CHAIN filename. Nie dotyczy to klucza /DE, dla którego domnianiem jest zawsze /NODE.

4. Dyrektywy

Dyrektywa musi być oddzielona od swoich argumentów i od komend MCR przez co najmniej jedną spację. Można wprowadzić dowolną liczbę spacji i tabulacji poziomej w trzech miejscach wiersza komendy:

- Na początku wiersza komendy
- Bezpośrednio po dwukropku (:) etykiety
- Na końcu wiersza komendy

Pozwala to na pisanie zbioru komend w formacie bardziej czytelnym. Zaleca się zaczynanie etykiety w kolumnie pierwszej, a wszystkiego innego w kolumnie dziewiątej /po znaku tabulacji

poziomej/. Ważnym wyjątkiem są wiersze pomiędzy dyrektywami .ENABLE i .DISABLE DATA. Żadne znaki puste, ani znaki tabulacji nie są usuwane z tych wierszy.

Przykład:

```
.IFT Z .GOTO 10
MAC @ASM
.10:   TKB @BLD
      .OPEN DATFIL
      .DATA XXXXX
      .ENABLE DATA
      TE DANE
      BEDA UMIESZCZONE
      W ZBIORZE DANYCH
      .DISABLE DATA
      .GOTO 20
```

5. Opis wybranych dyrektyw

.ETYKIETA: - definiuj etykietę

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Etykiety pojawiają się zawsze na początku wiersza. Mogą one występować w wierszu wraz z dodatkowymi dyrektywami i/lub komendą MCR, z komentarzem, bądź same. Gdy sterowanie przejdzie do wiersza z etykietą, następuje przetwarzanie tego wiersza poczynając od pierwszego znaku po dwukropku. Komendy nie muszą być oddzielone od etykiety spacją. W wierszu dopuszczalna jest tylko jedna etykieta. Etykiety mają długość 1 do 6 znaków i muszą być poprzedzone kropką, a kończone dwukropkiem. Etykiety mogą zawierać tylko znaki alfanumeryczne i/lub znak dolara (\$).

Przykład:

```
.100: .ASK A CZY CHCESZ KONTYNUOWAC?
      .IFT A .GOSUB 200
      :
      :
      :
      .200:
      ;TU JEST POCZATEK PODPROGRAMU
```

⋮
RETURN

.ASK - zadaj pytanie i czekaj na odpowiedź
XX
Dyrektywa .ASK wyprowadza pytanie na terminalu, czeka na odpowiedź i ustawia wyspecyfikowany symbol na wartość "prawda" bądź "fałsz", zależnie od otrzymanej odpowiedzi.

Format:

.ASK ssssss txt-strng

gdzie:

ssssss - Symbol o długości 1 do 6 znaków, któremu ma być przypisana wartość "prawda" lub "fałsz".

txt-strng - Dowolny ciąg znaków ASCII, poprzedzony co najmniej jednym znakiem pustym. Maksymalna długość ciągu wynosi 70 znaków.

Przy wyprowadzeniu ciągu znaków, procesor poprzedza go gwiazdką, a na końcu daje "? [Y/N]:" (tak/nie). Procesor rozpoznaje cztery odpowiedzi:

1. Y<CR> - ustaw symbol ssssss na "prawde".
2. N<CR> - ustaw symbol ssssss na "fałsz".
3. <CR> - ustaw symbol na "fałsz". <CR> oznacza znak powrotu karetki.
4. <ESC> - ustaw symbol ssssss na prawdę i ustaw logiczny symbol specjalny <ESCAPE> na prawdę tylko jeśli dozwolone jest rozpoznawanie <ESC>.

Przykład:

Wiersz dyrektywy:

.ASK A CZY CHCESZ ZAINSTALOWAC PIP
powoduje wyprowadzenie:

>* CZY CHCESZ ZAINSTALOWAC PIP? [Y/N]:
na terminalu. Symbol A zostaje ustawiony na "prawda" względnie "fałsz", gdy operator nacisnie Y (tak), N (nie), <CR>(powrót karetki) lub <ESC>(jeśli dozwolone jest rozpoznawanie <ESC>).

domniemania wartość zero. W tym przypadku granice, jeśli są wyszczególnione, muszą zawierać zero.

Odpowiedzią może być liczba oktalna lub dziesiętna. Poprzedzający liczbę znak (#) wskazuje liczbę oktalną, a kropka kończąca wskazuje liczbę dziesiętną. Jeśli nie ma żadnego znaku procesor przyjmuje typ domniemany. Domniemanym typem jest typ dziesiętny, gdy albo granice, albo wartość domniemana jest wyrażeniem dziesiętnym. W przeciwnym przypadku typem domniemany jest typ oktalny.

Przykłady:

Wiersz dyrektywy

```
.ASKN SYM ZDEFINIUJ SYMBOL LICZBOWY A
```

powoduje wyprowadzenie:

```
>*ZDEFINIUJ SYMBOL LICZBOWY A [O]:
```

na terminalu wydającym dyrektywę, gdzie:

[O] = domniemany typ (oktalnie).

Procesor definiuje symbol SYM zgodnie w wprowadzonej odpowiedzi.

Wiersz dyrektywy

```
.ASKN [2:35:16] NUMSYM ZDEFINIUJ SYMBOL LICZBOWY A
```

powoduje wyprowadzenie:

```
>*ZDEFINIUJ SYMBOL LICZBOWY A [O R:-35 D:16]:
```

w formacie [x R:low:hish D:def]

gdzie:

x - O, gdy domniemanym typem jest typ oktalny lub D, gdy dziesiętny.

R:low-hish - określa granice.

D:def - określa domniemanie.

Procesor sprawdza, czy odpowiedź znajduje się wewnątrz określonych granic.

Wiersz dyrektywy

```
.ASKN [NUMSYM+10:45NUMSYM+10] SYM ZDEFINIUJ  
SYMBOL LICZBOWY B
```

powoduje wyprowadzenie /przyjmując domniemaną wartość NUMSYM jako 16/

```
>*ZDEFINIUJ SYMBOL LICZBOWY B [O R:26-45 D:26]:
```


.ASKS - pytaj o definicje symbolu tekstowego

Dyrektywa **.ASKS** wyprowadza na terminalu zlecenie wprowadzenia tekstu dla zdefiniowania wyspecyfikowanego symbolu i opcjonalnie testuje, czy ilość znaków podanych w odpowiedzi mieści się wewnątrz określonych granic.

Format:

.ASKS ssssss txt-strng

.ASKS [low:hish] ssssss txt-strng
(Nawiasy są częścią składni)

gdzie:

ssssss - Symbol o długości 1 do 6 znaków, któremu przypisywana jest wartość tekstu.

txt-strng - Dowolny ciąg znaków ASCII poprzedzony przynajmniej jedną spacją. Ten ciąg znaków jest wyprowadzony przez procesor zbiorów komend. Maksymalna długość ciągu wynosi 70 znaków.

low:hish - Granice, w jakich musi być zawarta ilość znaków podana w odpowiedzi.

Wiersz komendy nie może przekraczać 80 znaków.

Przykłady:

Wiersz dyrektywy:

.ASKS NAM PODAJ SWOJE NAZWISKO

powoduje wyprowadzenie:

>#PODAJ SWOJE NAZWISKO [S]:

na terminalu, z którego wydano komendę. Procesor definiuje symbol NAM zgodnie z tekstem podanym w odpowiedzi.

Wiersz dyrektywy

.ASKS [1:15] IMIE PODAJ SWOJE IMIE

powoduje wyprowadzenie:

># PODAJ SWOJE IMIE [S R:1-15]:

w formacie [S R:low-hish]

gdzie:

S - typ symbolu /tekst/

R:low-hish - granice na ilość znaków

.CLOSE - zamyka zbiór wtórny

XX

Dyrektywa **.CLOSE** zamyka zbiór wtórny otwarty dyrektywą **.OPEN**.

Format:

.CLOSE [**#n**]

/nawiasy nie są częścią składni/

gdzie:

#n - opcjonalny numer zbioru w granicach 0-3 włącznie. Domniemanym jest **#0**. Można go zastąpić symbolem liczbowym ujętym w apostrofy.

.DATA - wprowadzanie danych do zbioru wtórnego

XX

Dyrektywa **.DATA** określa tekst, który ma być wprowadzany do zbioru wtórnego otwartego wcześniej dyrektywą **.OPEN**.

Format:

.DATA [**#n**] text-strng

/Nawiasy nie są częścią składni/

gdzie:

text-strng - tekst, który ma być wprowadzony do zbioru wtórnego.

#n - opcjonalny numer zbioru w granicach 0-3 włącznie. Domnianiem jest 0. Może być zastąpiony przez symbol liczbowy ujęty w apostrofy.

Wiersz komendy nie może być dłuższy niż 80 znaków. /Wyszczególniony tekst nie może być kontynuowany w następnym wierszu/. Jeśli zbiór wtórny nie jest otwarty, sygnalizowany jest błąd. Procesor wypisuje komunikat o błędzie i rozpoczyna przetwarzanie błędu.

Przykład:

.SET SEND "TO SA DANE"

.OPEN TEMP

.DATA 'SEND'

.CLOSE

Dyrektywy te wprowadzają TO SA DANE do zbioru wtórnego TEMP.DAT.

.DEC - zmniejsz wartość symbolu liczbowego
XX
Dyrektywa .DEC zmniejsza wartość symbolu liczbowego o 1.

Format:

.DEC ssssss

gdzie

ssssss - symbol liczbowy zawierający 1 do sześciu znaków.

Przykład:

.DEC N

Ta dyrektywa zmniejsza o 1 wartość przypisaną symbolowi liczbowemu N.

.DELAY - opóźnij wykonanie o wyspecyfikowany okres czasu

XX

Dyrektywa .DELAY opóźnia dalsze przetwarzanie zbioru komend o wyspecyfikowany okres czasu.

Format:

.DELAY nnu

gdzie:

nn - ilość jednostek czasu opóźnienia

u - T - takty

S - sekundy

M - minuty

H - godziny

Parametr nn jest przez domniemanie oktalny, a jeśli zostanie zakończony kropką, to dziesiętny.

Przy wykonaniu dyrektywy opóźnienia procesor wyprowadza komunikat:

AT.-- DELAYING

Gdy podany okres czasu upłynie, i zostanie podjęte wykonanie zadania, procesor wyprowadza komunikat:

AT.-- CONTINUING

Przykład:

Dyrektywa:

.DELAY 20M.

opóźnia przetwarzanie o 20/8/, czyli 16/10/ minut.

.DISABLE - blokowanie opcji

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Dyrektywa **.DISABLE** blokuje wyspecyfikowany tryb operacji (podstawiania, wyprowadzania danych, definiowania symboli globalnych, konwersji znaków dolnego rejestru lub rozpoznawania znaku <ESC> wcześniej aktywowanych dyrektywą **.ENABLE**.
Formaty:

.DISABLE SUBSTITUTION

.DISABLE DATA [#n]

/Nawiasy nie są częścią składni/

gdzie:

#n = opcjonalny numer zbioru w granicach 0-3.
Domniemaniem jest 0. Można zamiast wartości n podstawić symbol liczbowy ujęty w apostrofy.

.DISABLE GLOBAL

.DISABLE LOWERCASE

.DISABLE ESCAPE

.ENABLE - odblokowanie opcji

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Dyrektywa **.ENABLE** używana jest w celu dopuszczenia jednego z trybów operacji: konwersji znaków dolnego rejestru, wyprowadzania danych, definiowania symboli globalnych, podstawiania lub rozpoznawania znaku <ESC>. Każdy tryb jest niezależny od innych. Wszystkie mogą być aktywowane równolegle. W momencie startu procesora zbioru komend, odblokowany jest tylko tryb konwersji znaków dolnego rejestru. Pozostałe tryby są zablokowane.

W trybie konwersji znaków dolnego rejestru, znaki wczytane z terminala w odpowiedzi na dyrektywę **.ASKS** są zapamiętywane w ciągu symboli bez konwersji do górnego rejestru. Reprezentacja znaków jest istotna, kiedy ciągi są porównywane, wtedy to dyrektywa **.IF** traktuje znaki dolnego i górnego rejestru jako różne. Jeśli odblokowany jest tryb wyprowadzania danych, procesor wyprowadza wiersze, które występują po wierszu **.ENABLE DATA** do zbioru wtórnego.

W trybie definiowania symboli globalnych symboli, których nazwy zaczynają się od znaku dolara (\$), definiowane są jako symbole globalne do wszystkich poziomów zbiorów komend. Tak zdefiniowany symbol jest rozpoznawany na każdym poziomie zbiorów komend. Symbole, których nazwy nie rozpoczynają się od znaku \$, są rozpoznawane tylko na tym poziomie, na jakim są deklarowane.

W trybie podstawiania, procesor podstawia ciąg w miejsce symbolu. Symbol musi zaczynać się i kończyć apostrofem ('SYMBOL'). Na przykład, jeśli symbol A ma przypisaną wartość THIS IS A TEST, każdy symbol A zamieniany będzie na THIS IS A TEST. Jeśli aktywny jest tryb podstawiania, procesor przeprowadza podstawianie w każdym wierszu przed szukaniem w nim dyrektyw i komend MCR.

Rozpoznawanie <ESC> dopuszcza podawanie znaku <ESC> jako odpowiedź na dyrektywę .ASK, .ASKN i .ASKS. Jeśli pojedynczy znak <ESC> jest odpowiedzią na pytanie, logiczny symbol specjalny <ESC> przyjmuje wartość prawda. Znak <ESC> może być użyty tylko jako bezpośrednia odpowiedź na pytanie. Jeśli co najmniej jeden znak poprzedza <ESC>, wyprowadzany jest komunikat o błędzie.

Formaty:

- .ENABLE LOWERCASE
- .ENABLE SUBSTITUTION
- .ENABLE DATA [#n]

/Nawiasy nie są częścią składni/

gdzie:

#n = opcjonalny numer zbioru w granicach 0-3.
Domniemaniem jest 0. Zamiast wartości n można podstawić symbol liczbowy ujęty w apostrofy.

- .ENABLE GLOBAL
- .ENABLE ESCAPE

Przykłady:

Tryb podstawiania:

```
.ENABLE SUBSTITUTION
.ASKS FIL PODAJ NAZWE ZBIORU ZRODLOWEGO
MAC 'FIL'='FIL'
```

Podczas przetwarzania takiego zbioru komend, na terminalu wyprowadzane są wiersze:

```
* PODAJ NAZWE ZBIORU ZRODLOWEGO [S]: TEST
>MAC TEST=TEST
```

Tryb konwersji znaków dolnego rejestru:

```
.ENABLE SUBSTITUTION
.ENABLE LOWERCASE
.ASKS A ZDEFINIUJ SYMBOL TEKSTOWY A
;'A'
```

Podczas przetwarzania takiego zbioru komend na terminalu wyprowadzane będą następujące wiersze:

```
>* ZDEFINIUJ SYMBOL TEKSTOWY A [S]:Wartość sym-
bolu
);Wartość symbolu
```

Tryb symboli globalnych:

W zbiorze komend BOB1 pojawiają się następujące dwa wiersze:

```
.ENABLE GLOBAL
.SETS %X "TEST"
```

Zbiór o nazwie BOB2.CMD zawiera następujące wiersze:

```
.ENABLE SUBSTITUTION
```

```
@BOB1
```

```
MAC %X',%X'='%X'
```

To po wywołaniu zbioru BOB2.CMD MCR wyprowadza:

```
>@BOB1
```

```
>MAC TEST,TEST=TEST
```

```
>@<EOF>
```

Tryb danych:

```
.OPEN SECFILE.DAT
.ENABLE DATA
```

⋮

```
.DISABLE DATA
```

Wiersze, które umieszczone są pomiędzy dyrektywami .ENABLE i .DISABLE są wpisywane do zbioru SECFIL.DAT.

Tryb rozpoznawania znaku <ESC>:
;IF YOU WANT A LIST OF OPTION, TYPE <ESC>
.ENABLE ESCAPE
.ASKS A ENTER OPTION
.IFT <ESCAPE> .GOTO LIST

⋮
.LIST: ;OPTIONS ARE: A (ADD), S (SUBTRACT),
M (MULTIPLY)

Jeśli w odpowiedzi na ENTER OPTION wciśnięty będzie klawisz <ESC> /który nie jest wyprowadzony/, na terminalu wyprowadzone zostaną następujące wiersze:

>;IF YOU WANT A LIST OF OPTION, TYPE <ESC>
>#ENTER OPTIONS [S]: <ESC>
>;OPTIONS ARE: A (ADD), S (SUBTRACT), M (MULTIPLY)

.GOSUB - wywołanie podprogramu

XX

Dyrektywa .GOSUB zapamiętuje aktualną pozycję w zbiorze komend i powoduje skok do etykiety. Etykieta wskazuje punkt wejścia do podprogramu, z którego powrót jest poprzez dyrektywę .RETURN.

Maksymalne zagnieżdżenie wywoływanych podprogramów jest 8.

Format:

.GOSUB label

gdzie label jest etykietą, która wskazuje pierwszy wiersz podprogramu, ale bez poprzedzającej kropki i kończącego etykietę dwukropka. Etykieta musi być poprzedzona co najmniej jedną spacją.

Przykłady

Dyrektywa

.GOSUB EVAL

przekazuje sterowanie do podprogramów zaetykietowanego .EVAL:.

.GOTO - skok do etykiety

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Dyrektywa **.GOTO** powoduje skok z jednego wiersza w zbiorze komend do drugiego. Wszystkie komendy pomiędzy dyrektywą **.GOTO** a wyspecyfikowaną etykietą zostają zignorowane. Dozwolone są skoki w przód i wstecz zbioru komend.

Format:

.GOTO label

gdzie:

LABEL jest nazwą etykiety, lecz bez poprzedzającej kropki i kończącego dwukropka. Etykieta musi być poprzedzona co najmniej jedną spacją.

Przykład:

Dyrektywa:

.GOTO 100

Przekazuje sterowanie do wiersza zawierającego etykietę **.100:**.

WARUNEK LOGICZNY (IF)

Szereg dyrektyw sprawdza warunki, jeśli wynik sprawdzenia jest "prawda", to następuje przetwarzanie pozostałej części komendy. Warunki logiczne mogą być łączone w złożone warunki logiczne przy użyciu dyrektywy **.AND** i **.OR**.

.IF - sprawdź czy symbol spełnia wyspecyfikowany warunek

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Dyrektywa **.IF** porównuje symbol liczbowy lub tekstowy z innym wyrażeniem tego samego typu w celu sprawdzenia, czy jeden lub kilka możliwych warunków jest spełnione. Jeśli warunek jest spełniony, procesor wykonuje pozostałą część wiersza komendy. Jeśli porównywany jest symbol tekstowy z wyrażeniem tekstowym, procesor zbioru komend porównuje wartości ASCII każdego znaku operandu /od lewej do prawej/. Porównywany operand jest większy, jeśli pierwszy różny znak ma większą wartość od odpowiedniego znaku drugiego operandu. Symbole liczbowe porównywane są według ich wartości.

Format:

.IF symbol relop expr

gdzie:

symbol = 1 do 6-znakowy symbol liczbowy lub tekstowy

relop = jeden z następujących operandów relacji:

EQ lub = - równe

NE lub <> - nierówne

GE lub >= - większe lub równe

LE lub <= - mniejsze lub równe

GT lub > - większe

LT lub < - mniejsze

expr = wyrażenie tego typu co symbol.

Przykłady:

.SETS X "A"

.SETS Y "a"

.IF X LT Y .GOTO 200

Wartość ASCII symbolu tekstowego X jest mniejsza od wartości ASCII symbolu tekstowego Y, a więc warunek jest spełniony. Sterowanie przekazywane jest do wiersza zawierającego etykietę .200:.

.SETN N1 2

.SETN N2 7

.IF N1<=N2 PIP/LI

Warunek jest spełniony (symbol liczbowy N1 jest mniejszy lub równy symbolowi N2), wykonywana jest komenda PIP.

.SETS S1 "AAb"

.SETS S2 "AA"

.SETS S3 "BBBB"

.IF S1>=S2+S3[1:1] .INC N

Ponieważ warunek, w którym symbol S1 jest większy bądź równy symbolowi S2 połączonemu z pierwszym znakiem symbolu tekstowego S3 (AAb>=AAB), jest spełniony procesor zwiększa symbol liczbowy N.

.IFT/IFF - sprawdź czy symbol ma wartość
"prawda" czy "fałsz"

XX

.IFT/IFF testuje, czy symbol ma wartość "prawda" czy "fałsz". Jeśli test daje wynik "prawda", to AT. przetwarza pozostałą część wiersza komendy.

Formaty:

.IFT ssssss

.IFF ssssss

gdzie:

ssssss = 1- do 6-znakowy symbol, który jest sprawdzany.

Co najmniej jedna spacja musi poprzedzać i następować po symbolu logicznym.

Przykłady:

.IFT A .GOTO 100

.IFF B .GOTO 200

.IFDF/.IFNDF - sprawdź czy symbol jest zdefiniowany czy niezdefiniowany

XX

Dyrektywa **.IFDF** lub **.IFNDF** testuje, czy symbol logiczny, liczbowy lub tekstowy został zdefiniowany (**.IFDF**), czy też nie został zdefiniowany (**.IFNDF**). Jeśli sprawdzenie daje wynik "prawda", to przetwarzana jest pozostała część wiersza komendy. Dyrektywa ta nie sprawdza wartości symbolu.

Formaty:

.IFDF ssssss

.IFNDF ssssss

gdzie:

ssssss = 1- do 6-znakowy symbol sprawdzany.

Co najmniej jedna spacja musi poprzedzać i następować po symbolu.

Przykłady:

.IFDF A .GOTO 100

.IFNDF A .ASK A DO YOU WANT TO SET TIME

.IFINS/.IFNINS - sprawdź czy zadanie zostało
zainstalowane czy nie

XX
Dyrektywa .IFINS lub .IFNINS testuje, czy za-
danie zostało zainstalowane (.IFIND), czy nie
zostało zainstalowane(.IFNINS) w systemie. Jeś-
li wynik testu jest "prawda", to przetwarzana
jest pozostała część wiersza komendy.

Formaty:

.IFINS tttttt
.IFNINS tttttt

gdzie:

tttttt = nazwa zadania.

Nazwa zadania tttttt musi być poprzedzona i
musi po niej występować co najmniej jedna spac-
ja.

Przykłady:

.IFINS PIP .GOTO 250
.IFNINS PIP INS [1,50]PIP

.IFACT/.IFNACT - sprawdź czy zadanie aktywne
czy nieaktywne

XX
Dyrektywa .IFACT lub .IFNACT testuje, czy za-
danie jest aktywne (.IFACT), bądź czy jest uś-
pione (.IFNACT). Jeśli wynik testu jest "praw-
da", to przetwarzana jest pozostała część wier-
sza komendy.

Formaty:

.IFACT tttttt
.IFNACT tttttt

gdzie:

tttttt = nazwa zadania.

Nazwa zadania tttttt musi być poprzedzona i
musi po niej następować co najmniej jedna spac-
ja.

Przykłady:

.IFACT RAPORT .GOTO 350
.IFNACT RAPORT RUN RAPORT

.IFLOA/.IFNLOA - sprawdź, czy drajwer jest
załadowany, czy nie

XX

Dyrektywa .IFLOA lub .IFNLOA sprawdza, czy draj-
wer jest załadowany (.IFLOA), czy niezaładowany
(.IFNLOA). Jeśli sprawdzenie da wynik pozytywny,
to wykonywana jest pozostała część wiersza ko-
mendy. Dyrektywa zakłada, że drajwery rezydent-
ne są załadowane.

Formaty:

.IFLOA ddn:

.IFNLOA ddn:

gdzie:

ddn: - jest drajwerem urządzenia.

Przykłady:

.IFLOA DK: .GOTO 250

.IFNLOA DK: LOA DK:

WARUNKI ZŁOŻONE

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Warunki mogą być łączone przy użyciu dyrektyw
.AND oraz .OR. Ponadto, jeśli w tym samym wier-
szu pojawia się więcej niż jedno IF bez oddzie-
lenia za pomocą dyrektywy .AND, to realizowana
jest operacja .AND. W tym przypadku komenda wy-
konywana jest tylko wtedy, jeśli wynik sprawdze-
nia wszystkich warunków jest "prawda". Operato-
ry łączone .AND i .OR muszą być poprzedzone co
najmniej jednym znakiem spacji i co najmniej
jedna spacja musi być umieszczona po nich.
Dyrektywa .AND ma wyższy priorytet niż dyrekty-
wa .OR.

.INC - zwiększ wartość symbolu liczbowego

XX

Dyrektywa .INC zwiększa wartość symbolu liczbo-
wego o jeden.

Format:

.INC ssssss

gdzie:

ssssss = 1 do 6-znakowy symbol liczbowy.

Przykład:

.INC N

Zwiększ o jeden wartość przypisania symbolowi liczbowemu N.

/ - definicja logicznego końca zbioru

XX

Dyrektywa logicznego końca zbioru (/) kończy przetwarzanie zbioru wyprowadzając informację:

>@<EOF>

Format:

/

jako pierwszy nie pusty znak w wierszu. Dyrektywa ta może być umieszczona w dowolnym miejscu zbioru do szybkiego zakończenia przetwarzania zbioru.

Przykład:

.ASK CONT CZY CHCESZ KONTYNUOWAC

.IFT CONT .GOTO 100

/

.100:

.ONERR - skocz do etykiety w razie wykrycia błędu

XX

Jeśli procesor wykryje jeden z następujących błędów:

- Zadanie niezainstalowane w systemie (.XQT, .WAIT, .IFACT, .IFNACT)
- Niezdefiniowany symbol
- Zła składnia (.XQT, .WAIT, .DELAY)
- Nerozpoznana komenda
- Błąd podstawienia ciągu znaków (wszystkie komendy)
- Błąd typu symbolu (.IF, .IFT, .IFF, .INC, .DEC)
- Przedefiniowanie typu symbolu na inny typ (.ASK, .ASKN, .ASKS, .SETT, .SETF, .SETN, .SETS)
- Błąd zbioru danych (.OPEN, .OPENA, .DATA, .CLOSE, pomiędzy .ENABLE DATA i .DISABLE DATA) to przekazuje sterowanie do wiersza zawierającego wyspecyfikowaną etykietę. Cecha ta pozwala

użytkownikowi kontrolować zakończenie przetwarzania zbioru komend w sposób uporządkowany. Należy zauważyć, że dyrektywa .ONERR stosuje się tylko do błędów opisanych powyżej. Błędy zwracane z zadania zewnętrznego dla AT. (na przykład błąd składniowy MCR) nie są przetwarzane dyrektywą .ONERR.

Format:

.ONERR label

Przy wykryciu błędu procesor przekazuje sterowanie do wiersza rozpoczynającego się do .label:. Dyrektywa .ONERR musi być wykonana zanim AT. znajdzie błąd. Jeśli dyrektywa ta jest wykonana, to po wystąpieniu błędu sterowanie przekazywane jest do tej etykiety aż do wykonania nowej dyrektywy .ONERR. Jeśli etykieta wyspecyfikowana przez dyrektywę .ONERR nie istnieje, a wystąpił błąd, to kończy się przetwarzanie zbioru komend. Przykłady:

.ONERR 100

Po wykryciu jednego z błędów opisanych powyżej sterowanie przekazywane jest do etykiety .100:.

.OPEN - otwórz wtórny zbiór wyjściowy
XX

Dyrektywa .OPEN otwiera wyspecyfikowany wtórny zbiór jako zbiór wyprowadzania informacji.

Format:

.OPEN [$\#n$] filename

/Nawiasy nie są częścią składni/

gdzie:

filename - Zbiór wtórny, który ma być otwarty jako zbiór wyprowadzania danych. Domniemanym typem jest .DAT.

$\#n$ - Opcjonalny numer zbioru w granicach 0 do 3 włącznie pozwalający na otwarcie czterech zbiorów danych równocześnie. Domniemaniem jest $\#0$. Może być zastąpiony przez symbol liczbowy ujęty w apostrofy.

Przykłady:

.OPEN SECOUT

Otwórz zbiór SECOUT.DAT jako zbiór wyprowadzania informacji.

.OPENA - otwórz wtórny zbiór wyjściowy
XX

Dyrektywa **.OPENA** otwiera zbiór wtórny i wszystkie następne wprowadzane dane dołącza do zbioru (zapisuje na końcu zbioru).

Format:

.OPENA [n] filename

/Nawiasy nie są częścią składni/

gdzie:

filename - Zbiór wtórny, który ma być otwarty z następnymi danymi dołączonego do niego. Domniemanym typem zbioru jest **.DAT**.

n - Opcjonalny numer zbioru w granicach 0 do 3 włącznie, pozwalający na otwarcie czterech zbiorów danych równolegle. Domniemaniem jest 0. Może być zastąpiony przez symbol liczbowy ujęty w apostrofy.

Przykład:

.OPENA SECOUT

Dyrektywa otwiera zbiór SECOUT jako zbiór wyjściowy i dołącza następne dane do niego.

.PAUSE - czekaj na działanie operatora
XX

Dyrektywa **.PAUSE** pozwala użytkownikowi przerywać przetwarzanie zbioru komend celem oczekiwania na podjęcie działania ze strony operatora. Dyrektywa ta powoduje, że zadanie procesora zbiorów komend zawiesza się. Operator może wówczas zrealizować pewne operacje, a następnie spowodować kontynuację zadania.

Format:

.PAUSE

Gdy zadanie procesora zbiorów komend zawiesi się, wyprowadza następujący komunikat na terminalu:

AT.--PAUSING. TO CONTINUE TYPE "RES tttttt"
/AT. w trakcie pauzowania. Celem kontynuowania
wprowadzić "RES tttttt"/

gdzie:

tttttt = Nazwa zadania procesora zbiorów komend.
Operator wprowadza wówczas następującą komendę
dla aktywowania zadania:

>RES tttttt

Procesor zbiorów komend wyprowadza wówczas:

AT. -- CONTINUING

/AT. - w trakcie kontynuowania/

i kontynuuje przetwarzanie od miejsca, w którym
nastąpiła pauza.

.RETURN - powrót z podprogramu

XX

Dyrektywa .RETURN oznacza koniec podprogramu i
powrót do ostatnio zapamiętanej pozycji w zbiorze
komend (do wiersza występującego po dyrektywie
.GOSUB inicjującej podprogram).

Format:

.RETURN

.SETN - przypisz symbolowi wartość liczbową

XX

Dyrektywa .SETN definiuje względnie zmienia wartość
liczbową wyspecyfikowanego symbolu. Procesor
wyprowadza komunikat o błędzie w przypadku,
jeśli symbol został uprzednio zdefiniowany jako
symbol logiczny lub tekstowy.

Format:

.SETN ssssss numexp

gdzie:

ssssss = 1- do 6-znakowy symbol liczbowy.

numexp = wyrażenie liczbowe.

Przy obliczaniu wartości liczbowej przypisanej
symbolowi można tworzyć wyrażenia liczbowe, które
są kombinacją symbolu liczbowego lub stałej
z innym symbolem liczbowym lub stałą. Przy tworzeniu
wyrażeń liczbowych niedozwolone są znaki
spacji i znaki tabulacji. Wartość obliczana
jest od lewej do prawej chyba, że używane są

nawiasy tworzące podwyrażenia, które obliczane są w pierwszej kolejności. Typ wyrażenia jest oktalny, jeśli wszystkie operandy są oktalne, w przeciwnym przypadku typ jest dziesiętny.

Przykład:

.SETN LICZBA 27

Dyrektywa ta przypisuje symbolowi LICZBA wartość 27 (oktalnie).

.SETN AL 3*(A2-5)

Dyrektywa przypisuje symbolowi liczbowemu AL wartość symbolu A2 minus 5 pomnożone przez 3.

.SETS - przypisz symbolowi wartość alfanumeryczną

Dyrektywa .SETS definiuje względnie zmienia wartość tekstową wyspecyfikowanego symbolu. Procesor wyprowadza komunikat o błędzie, jeśli symbol został uprzednio zdefiniowany jako logiczny bądź liczbowy.

Format:

.SETS ssssss strexp

gdzie:

ssssss = 1- do 6-znakowy symbol tekstowy.

strexp = wyrażenie tekstowe.

Procesor przypisuje wyspecyfikowanemu symbolowi wartość tekstową obliczoną z wyrażenia tekstowego strexp. Jeśli w wyrażeniu używana jest stała tekstowa, musi być ona zamknięta w cudzysłowy ("strexp"). Wyrażenie tekstowe tworzone jest jako konkatenacja symboli tekstowych, stałych lub fragmentów pooddzielanych operatorem konkatenacji tekstów (+).

Przykłady:

.SETS A "ABCDEF"

Dyrektywa przypisuje symbolowi tekstowemu A wartość tekstu ABCDEF.

.SETS STR2 "ZZZ"

Dyrektywa przypisuje symbolowi tekstowemu STR2 wartość tekstu ZZZ.

.SETS X STR2+"ABC"

Dyrektywa przypisuje symbolowi tekstowemu X wartość symbolu STR2 plus ABC (to jest ZZZABC).

.SETS X STR2+A[1:3]

Dyrektywa daje wynik taki, jak poprzednia dyrektywa. Przypisuje symbolowi X wartość tekstową STR2 plus trzy znaki wartości symbolu A (to jest ZZZABC)

.SETS MYFILE <UIC>+"MYFILE.TXT"

Dyrektywa przypisuje symbolowi MYFILE wartość tekstu aktualnego UIC plus tekst MYFILE.TXT (np. gdy aktualny UIC jest [303,3], symbolowi tekstowemu MYFILE przypisana będzie wartość tekstu [303,3]MYFILE.TXT).

.SETT/.SETF - przypisz symbolowi wartość "prawda" lub "fałsz"

XX

Dyrektywy .SETT/.SETF definiują względnie zmieniają wartość wyspecyfikowanego symbolu logicznego. Procesor wyprowadza komunikat o błędzie, jeśli symbol został uprzednio zdefiniowany jako liczbowy bądź tekstowy.

Format:

.SETT ssssss

.SETF ssssss

gdzie:

ssssss - 1- do 6-znakowy symbol liczbowy. Symbol ssssss musi być poprzedzony jednym znakiem spacji.

Przykłady:

.SETT X

Dyrektywa przypisuje symbolowi X wartość prawda.

.SETF ABCDE

Dyrektywa przypisuje symbolowi ABCDE wartość fałsz.

.TEST - testuj symbol tekstowy

XX

Dyrektywa .TEST wstawia długość symbolu tekstowego do liczbowego symbolu specjalnego <STRLEN>.

Sprawdza on także znaki tekstu i ustawia logiczne symbole specjalne <ALPHAN> i <RAD50>

Format:

.TEST strsym

gdzie strsym jest testowanym symbolem tekstowym.

Przykład: Dyrektywa

.TEST A

wstawia ilość znaków tekstu A do <STRLEN> i odpowiednio ustawia <ALPHAN> i <RAD50>. Liczbowy symbol specjalny <STRLEN> może być następnie używany do porównania długości symbolu tekstowego A ze stałą liczbową lub wyrażeniem.

.WAIT - oczekuje na zakończenie zadania

XX

Dyrektywa .WAIT zawiesza przetwarzanie zbioru komend aż do zakończenia wyspecyfikowanego zadania.

Format:

.WAIT tttttt

Nazwa zadania tttttt musi być poprzedzona co najmniej jednym znakiem spacji.

Jeśli nazwa zadania jest pominięta procesor zbioru komend przyjmuje nazwę jaka wystąpiła w ostatniej komendzie MCR "RUN taskname". Nazwa ta specyfikowana jest jako

TTnn

gdzie:

TT = terminal wywołujący

nn = numer terminala

Jeśli określone /lub domniemane/ zadanie nie jest zainstalowane, dyrektywa .WAIT jest ignorowana.

Przykład:

.WAIT PIP

Dyrektywa zawiesza przetwarzanie zbioru komend aż zadanie PIP inicjowane z tego terminala zakończy się.

~~XX~~
.XQT - inicjuje równoległe wykonywanie zadań
Processor zbioru komend przekazuje komendę do MCR i oczekuje na zakończenie wykonywania tej komendy. Jednakże istnieje możliwość, by procesor aktywował jakieś zadanie i nie czekając na jego zakończenie, przystąpił do wykonywania następnej komendy zbioru komend. Komenda RUN MCR uruchamia zadania i procesor zbioru komend może kontynuować swe działanie natychmiast po przetworzeniu tej komendy. Natomiast nie można stosując komendę RUN przekazać wiersza komendy do zadania. Dyrektywa .XQT pozwala użytkownikowi uruchomić zadanie, przekazać wiersz komendy do niego i kontynuować przetwarzanie wierszy komend równoległe z uruchomionym zadaniem. AT. dopuszcza do 10/10/ kolejnych dyrektyw .XQT.
Format:

.XQT taskname comand

gdzie:

taskname = jest nazwą zadania /na przykład MAC, PIP, DMO/.

comand = jest komendą dla zadania.

Komenda .XQT pozwala na równoległe przetwarzanie zadań. Celem zsynchronizowania wykonania zadań równoległych, może użyć dyrektywy .WAIT w celu zawieszenia przetwarzania zbioru komend aż do zakończenia wyspecyfikowanego zadania.

Przykład:

.XQT MAC TEST,TEST=TEST

.XQT TKB BLD,BLD=BLD

.WAIT MAC

.WAIT TKB

Przykład uruchamiania asembler i budowanie zadania równoległe, a następnie czeka na zakończenie tych zadań.

KOMUNIKACJA Z UŻYTKOWNIKIEM

Niniejszy rozdział wyjaśnia czytelnikowi podstawowe koncepcje systemu operacyjnego DOC PB, które należy poznać przed rozpoczęciem współpracy z systemem operacyjnym.

1. Zadanie

Podstawową jednostką wykonywalną w systemie DOC PB jest zadanie. Zadanie może być uprzywilejowane lub nieuprzywilejowane. Zadania uprzywilejowane posiadają dodatkowe możliwości wykorzystania pamięci i urządzeń, których nie mają zadania nieuprzywilejowane, ale wykonywanie ich jest bardziej ryzykowne podczas działania systemu.

2. Partycje

Zadanie wykonuje się we wstępie wyznaczonym, ciągłym obszarze pamięci, nazywanym partycją. Partycja ma następujące parametry:

1. nazwa
2. określony rozmiar
3. stały adres początkowy
4. określony typ

Zależność pomiędzy zadaniem a partycją, w którym ono wykonuje się zależy od tego, czy system jest mapowany czy też niemapowany. W systemach niemapowanych, zadanie może wykonywać się tylko w partycji, która ma ten sam adres początkowy co partycja, dla której został utworzony obraz tego zadania za pomocą budowniczego zadań. Natomiast w systemach mapowanych, zadanie może być uruchomione w dowolnej partycji, byleby była ona dostatecznie duża, aby pomieścić to zadania.

2.1. Typy partycji

System operacyjny DOC PB współpracuje z dwoma zasadniczymi typami partycji:

1. Partycja zarządzana przez system
2. Partycja zarządzana przez użytkownika

W partycji zarządzanej przez system, egzekutor przydziela dostępny obszar pamięci tak, aby pomieścić w partycji możliwie dużą liczbę zadań w danej chwili. Przydział taki może powodować kompresje miejsca zajmowanego przez rezydujące zadania, aby z wolnego miejsca utworzyć ciągly blok, na tyle duży, by pomieścił zleczone zadanie. Partycje zarządzane przez system mogą wystąpić tylko w systemach mapowanych.

Partycja zarządzana przez użytkownika jest wyłącznie przydzielona jednemu zadaniu w danym czasie. Partycja tego typu może wystąpić zarówno w systemie mapowanym jak i w systemie niemapowanym.

W obydwu typach partycji egzekutor może chwilowo wyładować na dysk rezydujące zadanie po to, aby utworzyć miejsce dla drugiego zadania.

2.2. Podpartycje

Partycja zarządzana przez użytkownika może być podzielona na maksymalnie 7 niezachodzących na siebie podpartycji. Podobnie jak jej macierzysta partycja główna, podpartycja może zawierać w danym czasie tylko jedno zadanie. Ze względu na to, że podpartycja zajmująca tę samą fizyczną pamięć co partycja główna, zadania nie mogą równocześnie rezydować zarówno w partycji głównej jak i w jednej lub więcej podpartycjach. Jednakże, ze względu na to, że każda podpartycja może zawierać jedno zadanie, w początkowo pustej partycji głównej można potencjalnie uruchomić równocześnie maksymalnie 7 zadań.

3. Wieloprogramowość

Zadania ubiegają się o zasoby systemu /w tym także o pamięć/ na bazie priorytetu i dostępności zasobów. Priorytet zadania określa liczba, która została przypisana do zadania w czasie,

gdy zadanie to było utworzone przez BUDOWNICZEGO ZADAN (TKB), albo gdy było instalowane. Liczba leży w zakresie 1 do 250 /dziesiętnie/, gdzie wyższa liczba oznacza wyższy priorytet. Zadanie mające najwyższy priorytet, ma dostęp do wszystkich zasobów, jakich potrzebuje i wykorzystuje procesor.

Często zadanie jest zablokowane w chwili, gdy wymaga obsługi przez system. Na przykład, może się zablokować w oczekiwaniu na zakończenie operacji wejścia/wyjścia. W czasie, gdy zadanie jest zablokowane w ten sposób, egzekutor sprawdza, czy jest jakieś inne zadanie, które chciałoby użyć procesora. Zostanie wybrane to zadanie, które ma najwyższy priorytet i dostęp do wszystkich zasobów, jakich potrzebuje.

4. Tworzenie zadania

Celem utworzenia zadania i przygotowania go do wykonywania, użytkownik systemu operacyjnego DOC PB wykonuje następujące cztery podstawowe kroki:

1. Wprowadza program w języku źródłowego przy pomocy edytora.
2. Transluje kod źródłowy przy pomocy odpowiedniego translatora, który wytwarza zbiór pośredni.
3. Buduje zadanie przy pomocy BUDOWNICZEGO ZADAN (TKB), który wytwarza zbiór obrazu zadania.
4. Wreszcie, wydaje komendę MCE celem zainstalowania zbioru obrazu zadania w systemie. Zbiory otrzymane z kroków 1, 2 i 3 są zapamiętywane w tomie o strukturze zbiorów DOC PB.

4.1. Instalowanie zadania

Gdy operator instaluje zadanie /wydając komendę MCR "INS", bądź jakąś postać komendy "RUN"/, system wprowadza parametry zadania do tablicy rezydującej w systemie, nazywanej katalogiem

zadań systemowych (STD). Parametry te zawierają nazwę i długość zadania oraz adres w tomie, który zawiera obraz zadania. Zainstalowane zadanie definiowane jest jako zadanie, które jest umieszczone w katalogu STD. Zadanie takie nie rezyduje w pamięci, ani nie zbiega się o zasoby systemu. System traktuje je jako uśpione, dopóki nie nastąpi zlecenie aktywowania go.

Zainstalowane, uśpione zadanie można aktywować przez wydanie komendy MCR "RUN", bądź z innego zadania poprzez jedną z dyrektyw egzekutora: RQSTX, RUNX lub SPWNX.

Gdy egzekutor otrzyma którąkolwiek z postaci zlecenia uruchomienia, realizuje szereg działań:

1. Przydziela niezbędne zasoby.
2. Powoduje, że to zadanie aktywnie ubiega się o zasoby systemu wraz z innymi zadaniami aktywnymi znajdującymi się w pamięci.
3. Ładuje zadanie do pamięci /jeśli w jego partycji jest wolne miejsce/.

UWAGA

Celem uzyskania miejsca dla zadania, egzekutor może wyładować przeładowywalne zadanie o niższym priorytecie, jeśli takowe istnieje. Jeśli partycja, w której zadanie jest zainstalowane, jest całkowicie zajęta, a żadne z rezydujących zadań nie może być wyładowane, zadanie zostaje umieszczone w kolejce innych aktywnych zadań, z których każde oczekuje na miejsce w swej partycji. Należy zwrócić uwagę na to, że liczba zainstalowanych, uśpionych zadań może znacznie przekraczać i zazwyczaj przekracza, liczbę zadań aktywnych.

WPROWADZENIE

EDI jest wierszowo zorientowanym edytorem przeznaczonym do tworzenia i modyfikacji zbiorów tekstowych. EDI działając na zbiorach tekstowych, zapisanych w kodzie ASCII, jest często używany do tworzenia i porządkowania zbiorów zawierających programy źródłowe pisane w języku FORTRAN lub MACRO.

Komendy EDI dzielą się na następujących siedem grup:

- * Komendy sterujące - wybierają warunki działania, zamykają i otwierają zbiory, wybierają tryby odbioru i przekazywania danych.
- * Komendy lokalizujące - sterują położeniem wiersza bieżącego i w ten sposób określają, który wiersz tekstu jest aktualnie poddawany działaniu.
- * Komendy modyfikacji tekstu - zmieniają zawartość wierszy tekstu.
- * Makrokomendy - definiują, przechowują wykonując wcześniej zdefiniowane sekwencje komend edytora wierszowego.
- * Komendy wprowadzania/wyprowadzania zbioru przesyłają tekst z i do zbioru wejściowego/wyjściowego oraz przechowują te zbiory.
- * Komendy wydruku - realizują wyprowadzanie tekstu na terminal lub drukarkę.
- * Komendy końca pracy - kończą działania redagujące.

1. Sposoby korzystania z edytora wierszowego

1.1. Metody wywołania EDI

W celu wywołania edytora wierszowego należy skorzystać jedną z metod wywoływania systemowych programów pomocniczych.

Jeżeli wykorzystywany jest którykolwiek z formatów, z wyjątkiem formatu ">EDI specyfikacja-zbioru", EDI wydaje swe przynaglenie w postaci:
EDI>

W tym miejscu należy wprowadzić specyfikację zbioru przeznaczanego do edycji.

1.1.1. Specyfikacja zbioru wejściowego

Specyfikacja zbioru wejściowego posiada następującą postać:

urz:[ufd]nazwa-zbioru.typ-zbioru;wersja
Jeżeli specyfikowany zbiór jest nowym zbiorem /ma to miejsce wtedy, gdy wyszczególniony zbiór nie został znaleziony na określonym urządzeniu/, EDI przyjmuje to jako zlecenie utworzenia nowego zbioru o podanej nazwie. EDI drukuje wówczas następujące wiersze komunikatu:

[CREATING NEW FILE]

INPUT

(Tworzenie nowego zbioru)

(Wprowadzenie)

i wchodzi w tryb wprowadzania. Jeśli wyprowadzony jest komunikat:

FILE DOES NOT EXIST

(Zbiór nie istnieje)

oznacza to, że nie istnieje katalog zbiorów użytkownika odpowiadający specyfikowanemu UIC. EDI nie przyjmuje specyfikacji zbioru komend. Jeśli wyszczególniona jest nazwa zbioru istniejącego, wtedy edytor wierszowy drukuje:

[OOOnn LINES READ IN]

[PAGE 0]

■

i czeka w trybie wydawania komend na wydanie pierwszej komendy.

Jeśli wykorzystany jest format wywołania edytora "EDI specyfikacja-zbioru", wówczas edytor nie wydaje przynaglenia (EDI) i rozpoczyna pracę w trybie wprowadzania lub wydawania komend, zależnie od specyfikowanej nazwy zbioru. Tryb wprowadzania jest ustawiany, jeśli nazwa zbioru jest nazwą nowego zbioru, a tryb wydawania komend, jeśli nazwa zbioru jest nazwą już istniejącego zbioru.

W chwili rozpoczęcia pracy, po zidentyfikowaniu zbioru wejściowego i założeniu zbioru wyjściowego, edytor jest gotowy do przyjmowania komend. W trybie wydawania komend, pierwszy dostępny do edycji wiersz znajduje się w miejscu poprzedzającym wiersz początkowy zbioru wejściowego lub bufora blokowego. Dzięki temu można wstawić tekst na początku zbioru wejściowego lub bufora blokowego wydając komendę INSERT. W celu manipulowania pierwszym wierszem tekstu, konieczne jest wykonanie operacji NEXT, aby wiersz ten stał się dostępny do przetwarzania.

1.2. Tryby sterujące: wprowadzania i wydawania komend

EDI może być uruchamiany w dwóch trybach:

* Tryb wydawania komend /tryb komend/.

* Tryb wprowadzania /tryb tekstu/.

Tryb wydawania komend jest przyjmowany automatycznie, wtedy gdy użytkownik wywołuje zbiór już istniejący.

W trybie wydawania komend EDI wyprowadza gwiazdkę (*), jako przynaglenie.

Tryb wprowadzania jest przyjmowany automatycznie na początku pracy programu, jeśli wyszczególniono zbiór nieistniejący.

Znajdując się w trybie wprowadzania edytor wierszowy nie ukazuje jawnego przynaglenia. Wiersze wprowadzane z klawiatury traktowane są jako tekst i umieszczone są w zbiorze wyjściowym. Gdy wejściowy wiersz zakończony jest powrotem karetki (CR), EDI realizuje na terminalu zakończenie i przejście do następnego wiersza.

W celu przejścia z trybu wydawania komend na tryb wprowadzania, należy wydać komendę INSERT, po której następuje powrót karetki. Aby powrócić do trybu wydawania komend, należy wprowadzić powrót karetki jako jedyny znak w wierszu wejściowym. EDI drukuje wówczas "*" jako przynaglenie, potwierdzające tryb wydawania komend.

1.3. Tryby dostępu do tekstu

EDI posiadają dwa tryby dostępu i operowania wierszami tekstu zbioru wejściowego /wiersz jest zdefiniowany jako ciąg znaków zakończony - nych znakiem powrotu karetki/. Trybami tymi są:

* Tryb wierszowy - pozwala w danej chwili na dostęp do jednego wiersza tekstu; niemożliwe jest cofanie się.

* Tryb blokowy - umożliwia dostęp do dowolnych wierszy wewnątrz bloku. Możliwe jest cofanie się wewnątrz bloku, natomiast cofanie się do bloku poprzedniego nie jest możliwe.

Tryb blokowy jest domniemanym trybem dostępu do tekstu.

Tablica 1 zawiera streszczenie różnic między trybem wierszowym a blokowym.

Tablica 1

Porównanie trybu wierszowego z blokowym.

Tryb wierszowy

Tryb blokowy

W danej chwili dostępny do modyfikacji jeden wiersz

W danej chwili dostępne do modyfikacji wszystkie wiersze wprowadzonego bloku.

Można uzyskać dostęp tylko do tych wierszy w zbiorze które znajdują się za wskaźnikiem wiersza bieżącego.

Możliwy jest dostęp zarówno do wierszy wewnątrz bloku znajdujących się za jak i przed wskaźnikiem wiersza bieżącego.

Komendy lokalizujące mogą być wykorzystane do szukania danego ciągu znaków w całym zbiorze wejściowym.

Komendy lokalizujące odnoszą się tylko do bloku znajdującego się aktualnie w pamięci. W celu przeszukania większej ilości danych należy wczytać następny blok.

1.4. Konwencje komend edytora

EDI używa gwiazdki (*) i wielokropka (...) w specjalny sposób, opisany w tym podrozdziale. Opisana jest także konwencja symboli używanych w celu skracania definicji komend edytora.

1.4.1. Wykorzystanie gwiazdki (*)

Znak gwiazdki (*) może być użyty w miejscu jakiegokolwiek argumentu liczbowego, przyjmującego wartość do 32767 /dziesiętnie/.
Przykład: Następująca komenda powoduje wyprowadzenie na terminal reszty bufora blokowego lub zbioru.

PRINT *

1.4.2. Wykorzystanie wielokropka (...) w miejscu szukanego tekstu

Przy niektórych komendach definiowany jest tekst, który ma być odszukany i/lub poprawiony. W celu skrócenia wprowadzania z terminala potrzebnych fragmentów tekstu wejściowego, w większości wypadków użytkownik może posłużyć się specjalną konstrukcją tego tekstu. W tych specjalnych przypadkach wielokropek (...) występuje zamiast jakiegokolwiek ciągu znaków z danego tekstu.

Przypadek 1.

tekst1...tekst2 Dowolny ciąg znaków, którego początek określony jest przez znaki "tekst1", a koniec określony przez pierwsze napotkanie znaków tekst2.

Przypadek 2.

...tekst Jakiś ciąg znaków, który rozpoczyna się od początku bieżącego wiersza, a kończy pierwszymi napotkanymi znakami oznaczonymi jako "tekst".

Przypadek 3.
tekst... Pierwszy ciąg znaków, który rozpoczą na się znakami "tekst", a kończy wraz z końcem bieżącego wiersza.

Przypadek 4.
... Cały bieżący wiersz wejściowy.

Przykłady:
W następujących przykładach, komenda CHANGE jest użyta z czterema przypadkami specjalnej konstrukcji tekstu podanymi powyżej. W każdym przypadku wskaźnik wiersza ustawiony jest na danym wierszu bieżącym:

THIS IS A SAMPLE OF SPECIAL STRING CONSTRUCTS

Przypadek 1.

C /S A...E O/S AN EXAMPLE O

Wynik działania:

THIS IS AN EXAMPLE OF SPECIAL STRING CONSTRUCTS

Przypadek 2.

C/...SPEC/THIS IS AN EXAMPLE OF SPEC

Wynik działania:

THIS IS AN EXAMPLE OF SPECIAL STRING CONSTRUCTS.

Przypadek 3.

C/STRING.../EDI STRING CONSTRUCTS

Wynik działania:

THIS IS AN EXAMPLE OF SPECIAL EDI STRING CONSTRUCTS.

Przypadek 4.

C/.../EXAMPLES OF SPECIAL EDI CONSTRUCTS.

Wynik działania:

EXAMPLES OF EDI CONSTRUCTS

1.4.3. Skróty komend

EDI dopuszcza wykorzystanie możliwości skracania komend. Gdy dozwolone są skrócone formy, specyfikacja formatu komendy przedstawiona jest wraz z dozwolonym skrótem dużych liter. Małe litery mogą być wprowadzone opcjonalnie. Następujący przykład obrazuje dopuszczalne skracanie komendy VERIFY. Specyfikacja formatu komendy jest następująca:

Verify
Dozwolone jest następujące skracanie komendy
VERIFY:

V

VE

VER

VERI

VERIF

VERIFY

WPROWADZENIE

1. Cele rejestracji błędów

Podsystem rejestrowania błędów sprzętu systemu operacyjnego DOC PB-2 kontroluje niezawodność sprzętu systemu. W sposób ciągły wykrywa on i rejestruje informacje o każdym błędzie pamięci dyskowej, pamięci na taśmie magnetycznej oraz każdy błąd parzystości pamięci operacyjnej, w miarę jak błędy takie powstają, niezależnie od tego, czy są one korygowalne czy też nie. Następnie, można uruchomić zadanie generujące sprawdzania o poszczególnych błędach i/lub zestawienia sprawozdawcze, dotyczące wszystkich lub niektórych spośród powyższych błędów. Nawet bez korzystania z rejestrowania błędów, drajwer automatycznie usiłuje skorygować błędy, które można skorygować. Po udanym skorygowaniu użytkownik może nie zdawać sobie sprawy z tego, że wystąpił błąd.

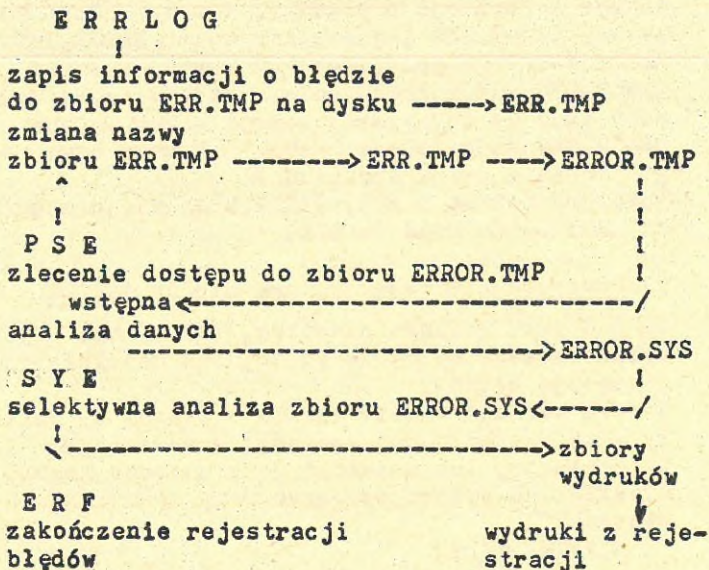
Jeśli sprawdzania o błędach wydają się wskazywać na rozszerzającą się wadę, zarządzający systemem powinien porozumieć się z przedstawicielem serwisu, który wykorzysta sprawozdania dla diagnozy problemu. Inżynier serwisu będzie mógł określić przyczynę szybciej, jeśli ma do dyspozycji sprawozdanie, które opisuje błędy, powstające tuż przed ostateczną awarią.

Rejestracja błędów realizuje następujące działania:

1. Wykrywa błędy sprzętowe w chwili ich powstania,
2. Gromadzi dane o błędzie i zapamiętuje je w zbiorze na dysku,
3. Formatuje te informacje w celu wytworzenia sprawozdania o błędach.

Rejestracją błędów sterują wspólnie procedury w egzekutorze i zadania rejestracji błędów. Zadaniami rejestracji błędów są: ERRLOG, PSE, SYE i ERF. Informacje o błędach gromadzone przez

procedury egzekutora (na przykład, stan rejestrów przy pojawieniu się błędu urządzenia), stanowią zapisy stanu systemu w czasie błędu. Poza wszelkimi informacjami systemowymi, procedury błędu identyfikują rodzaj błędu i związane z nim urządzenie, rejestrują czas powstania błędów i przypisują kolejny numer każdemu błędowi. Główna funkcja zadań rejestrowania błędów jest przechowywanie informacji o błędach, zgromadzonych przez egzekutor, w celu wytworzenia sprawozdania. ERRLOG chroni informacje z egzekutora i przekazuje je do PSE w chwili uruchomienia tego zadania. PSE przygotowuje dane do wprowadzenia do zadania SYE, które selektywnie generuje końcowe sprawozdanie w czytelnej postaci. Poniższy rysunek przedstawia współdziałanie zbiorów.



Rys.1. Współdziałanie zadań.

Za każdym razem, gdy zostanie zainicjowane zadanie ERRLOG (zwykle przy uruchomieniu systemu), tworzy ono zbiór nazywany ERR.TMP i zapisuje w nim informacje o konfiguracji systemu, opcjonalnie potrzebne dla zadania PSE. ERRLOG zapisuje następnie okresowo do tego zbioru komunikaty błędów, które otrzymuje z kolejki organizowanej przez egzekutor. Przez domniemanie, dane te przekazywane są do zbioru SYO:[1,6] ERR.TMP. [1,6] i ERR.TMP są parametrami przyjmowanymi z założenia. Gdy użytkownik uruchomi PSE, ERRLOG zmienia nazwę zbioru danych na ERROR.TMP, czyniąc je dostępne dla PSE, a następnie tworzy nowy zbiór ERR.TMP, który zawiera potrzebne informacje o konfiguracji systemu. W danym momencie może istnieć tylko jeden zbiór ERR.TMP. Zadanie wstępnego formatowania PSE wymaga aby zadanie ERRLOG utworzyło zbiór ERROR.TMP, formatuje informacje o błędach, tworzy zbiór nazywany ERROR.SYS i następnie usuwa ERROR.TMP. Jeśli istnieje już jeden lub więcej zbiorów ERROR.SYS, PSE dołącza nowe, wstępnie sformatowane dane do najwyższej wersji zbioru. Dane w ERROR.SYS mają format, który SYE może użyć do generowania swych sprawozdań. Zbiór wyjściowy z PSE, ERROR.SYS, staje się zbiorem wejściowym do SYE.

2. Uruchamianie rejestratora błędów ERRLOG

ERRLOG jest jedynym zadaniem, które musi być zainstalowane na stałe po to, aby działała rejestracja błędów.

W celu zainstalowania ERRLOG w systemie, należy użyć następującej komendy, wprowadzając ją z terminala, lub umieścić ją w zbiorze komend STARTUP.CMD wykorzystywanym przy uruchamianiu systemu:

```
MCR>INS MERL
```

Zbiór ERL.TSK zawiera obraz zadania ERRLOG.

Celem rozpoczęcia rejestracji błędów, należy zainicjować ERRLOG z uprzywilejowanego terminala wprowadzając komendę:

```
>RUN ERRLOG
```

Powyższa komenda może znajdować się w zbiorze komend startu systemu (LB:[1,2]STARTUP.CMD). W tym przypadku, rejestrowanie błędów zostało uruchomione automatycznie przy startowaniu systemu.

Należy zauważyć, że ERRLOG wymaga, by katalog [1,6] istniał na dysku systemowym, natomiast zadanie PSE wykonywane jako nieprzywilejowane musi mieć pełny dostęp do tego katalogu (RWED).

3. Uruchamianie wstępnego formatera PSE

W celu zainstalowania i uruchomienia PSE w systemie, należy wprowadzić komendy:

```
>INS X PSE
```

```
>PSE
```

Zadanie wstępnie formatujące zgłasza się przyznaczeniem PSE) i oczekuje, by użytkownik wprowadził linie komendy:

Wiersz komendy zadania PSE ma postać:

```
urz.wyj:[katalog]zbiór.typ=urz.wej:
```

gdzie wyjściowy zbiór opisany jest przez standardową specyfikację zbioru w systemie DOC PB-2. Specyfikacja zbioru wejściowego (urz.wej:) składa się tylko ze specyfikacji urządzenia wejściowego, gdyż nazwa zbioru (przydzielana przez zadanie ERRLOG) brzmi zawsze ERROR.TMP i zawsze stosowany jest katalog UFD [1,6].

Zarówno wywołanie dla PSE jak i specyfikacje zbiorów można umieścić w tym samym wierszu, czyli:

```
>INS X PSE
```

```
>PSE urz.wyj:[katalog]zbiór.typ=urz.wej:
```

Domniemania dla pominiętych pól specyfikacji zbioru wyjściowego są następujące:

Pole	Domniemanie
urz.wyj:	SYO:

katalog [1,6]
zbiór ERROR
typ SYS
urz.wej: SYO:

W celu użycia wszystkich domniemanych wartości zarówno dla specyfikacji zbiorów wejściowych, jak i wyjściowych, należy nacisnąć klawisz <CR w odpowiedzi na przynaglenie PSE>. Gdy PSE przynagli ponownie, należy wprowadzić CTRL/Z celem powrotu do MCR.

Przykład:

```
>INS X PSE ;instalowanie PSE  
>PSE ;uruchomienie PSE  
PSE> SY:=SY: ;generowanie zbioru wstępnie  
sformatowanego  
PSE>^Z ;wyjście z PSE  
>REM PSE ;usunięcie PSE
```

Zadanie PSE nie może być uruchomione przy pomocy komendy RUNK. Zadanie to musi być zainstalowane z uprzywilejowanego terminala i uruchomione przy pomocy komendy PSE <CR>.

4. Uruchomienie generatora sprawozdań SYE

Istnieją dwa sposoby uruchomienia zadania generatora sprawozdań SYE.

1. Wprowadzenie komendy z terminala uprzywilejowanego:

```
>INS XSYE  
>SYE
```

2. Wprowadzenie komendy z dowolnego terminala:

```
>RUN XSYE
```

Zadanie generatora sprawozdań odpowiada komunikatem i przynagleniem SYE), po czym czeka, by użytkownik wprowadził wiersz komendy.

Na przykład:

```
>RUN XSYE  
SYE>
```

4.1. Wiersz komendy zadania SYE

Format wiersza komendy jest następujący:

urz.wyj:[katalog]zbiór.typ=urz.wej:[katalog]

zbiór.typ/klucz1.../klucz2

Zbiór wejściowy opisany jest przez standardową specyfikację zbiorów systemu DOC PB-2 musi to być zbiór utworzony uprzednio jako zbiór wyjściowy przez PSE.

Domniemane wartości przy określaniu zbioru wyjściowego są następujące:

Pole	Domniemanie
------	-------------

urz.wyj:	SYO:
katalog	UIC użytkownika
zbiór	ERRLOG
typ	LST

Domniemane wartości przy określaniu zbioru wejściowego są następujące:

Pole	Domniemanie
------	-------------

urz.wej	SYO:
katalog	[1,6]
zbiór	ERROR
typ	SYS
klucze	/-RP/SP

W celu wykorzystania wszystkich domniemań, zarówno dla specyfikacji zbioru wejściowego, jak i wyjściowego należy, w odpowiedzi na przynaglenie SYE, nacisnąć klawisz <CR>. GDY SYE przynagli ponownie, należy wprowadzić CTRL/Z celem powrotu do MCR.

Na przykład:

SYE)<CR>

SYE)<Z>

Przy określaniu rodzaju sprawozdań z rejestracji błędów, do specyfikacji zbioru wejściowego można dołączyć następujące klucze:

KLUCZ

OPIS

/RP:[KLASA] klucz RP tworzy sprawozdanie o poszczególnych błędach; parametr KLASA wyznacza rodzaje błędów, któ-

re mają być raportowane. Gdy klucz ten pomija KLASE, SYE raportuje wszystkie klasy błędów.

Klasa ma jedną z następujących wartości:

pusta wszystkie klasy

HDW wszystkie błędy sprzętowe. Gdy KLASA określona jest jako HDW, może wystąpić dodatkowo parametr TYP, definiujący typ błędów sprzętowych, które będą uwzględniane w raporcie. Parametr ten podaje się w następującej postaci:

/RP:HDW[:TYP]

gdzie TYP może przyjmować następujące wartości:

pusty wszystkie rodzaje błędów sprzętowych

DSK wszystkie błędy dyskowe

MAG wszystkie błędy taśmy

MEM wszystkie błędy parzystości pamięci operacyjnej i buforowej

SYS[:TYP] wszystkie błędy systemowe (startowania rejestracji błędów, wyłączania rejestracji oraz błędy PSE), które są istotne dla serwisu producenta.

TYP może przyjmować następujące wartości:

pusty wszystkie błędy systemowe

PSE wszystkie błędy związane z uruchomieniem zadania wstępnie formatującego PSE

STA wszystkie błędy związane ze startowaniem rejestracji błędów

TIM błędy związane ze zmianą czasu systemowego

MSG błędy związane z obsługą komunikatów systemowych

TMO wszystkie błędy polegające na wyczerpaniu się czasu oczekiwania na przerwanie.

/RP=TMO[:TYP]

gdzie:

TYP może przyjmować następujące wartości:

DSK wszystkie błędy pamięci dyskowej

MAG wszystkie błędy na taśmie magnetycznej

/-RP lub /NORP - nie podawać w sprawozdaniu opisu poszczególnych błędów. Zadanie SYE wytwarza tylko zestawienie sprawozdawcze. Domniemaniem jest /-RP.

/SU podaje w sprawozdaniu podsumowujące zestawienie informacji

/QU podaje skrócone sprawozdanie

/DV:URZADZENIE[N] - podanie w sprawozdaniu tylko tych błędów, które pojawiły się w podanym typie urządzeń bądź jednostce urządzenia.

Oprócz standardowych typów urządzeń można podać:

CMM - urządzenie transmisji danych

UDI - niezdefiniowane przerwanie

PWR - zaniki zasilania

Przykłady:

/DV:DK zleca podania statystyki wszystkich błędów związanych z wszystkimi jednostkami dysków SMRK.

/DV:DK1 zleca podania statystyki błędów tylko dla jednostki 1 dysku SMRK.

Pozostałe klucze pozwalają użytkownikowi modyfikować szczegółową zawartość sprawozdania:

KLUCZ

OPIS

/BEG:DATA-I-CZAS - w sprawozdaniu podać tylko te błędy, które wystąpiły nie wcześniej niż w określonym czasie i dacie.

Parametr data-i-czas ma następującą postać:

dd-mon-yr:hh:mm:ss

gdzie wszystkie liczby są dwucyfrowe dziesiętne i trzeba określić wszystkie sześć pól.

Przykład:

/BEG:15-NOV-83:20:30:00 spowoduje podanie tych błędów, które powstały po godzinie 20:30 15 listopada 1983 r. Domniemaniem jest podanie wszystkich błędów podanego typu, bez względu na to, kiedy powstały.

/END:DATA-I-CZAS - objęcie sprawozdaniem tylko tych błędów, które powstały nie później niż w określonym czasie i dacie. Format czasu i daty jest następujący:

dd-mon-yr:mm:ss

gdzie wszystkie liczby są dwucyfrowe dziesiętne i trzeba wyspecyfikować wszystkie pola.

Przykład:

/END:18-OCT-83:18:00:00 powoduje podanie wszystkich błędów, które pojawiły się przed godziną 18:00, 18 października 1983.

/FUL[L] Wytworzenie szczegółowych indywidualnych sprawozdań, które obejmują następujące informacje o każdym błędzie:

- zawartość rejestrów urzędzenia
- indentyfikacja jednostki urzędzenia
- liczba wykonanych powtórzeń operacji
- błąd skorygowany/nieskorygowany
- opis bitów rejestru błędów urzędzenia
- zadanie i funkcja, które wywołały błąd
- informacja o bieżącym działaniu magistrali

Z założenia przyjmowane jest /-FUL bądź /NOFUL. Sprawozdanie nie obejmuje wtedy ostatnich trzech pozycji informacji wymienionych powyżej.

/HEL[CP] klucz /HELP powoduje, że zadanie SYE wyprowadza instrukcje rejestrowania błędów. W celu otrzymania tej informacji, użytkownik specyfikuje klucz /HELP w swej odpowiedzi na przynaglenie SYE, w następującej postaci:
SYE)/HELP<CR>

/SP wyprowadza zbiór wyjściowy metodą systemowego wyprowadzania

5. Uruchamianie zadania wyłączania rejestracji ERF

W celu zakończenia rejestracji błędów należy uruchomić zadanie ERF wprowadzając poniższe z uprzywilejowanego terminala:

>INS XERF

>ERF

Komenda RUN XERF nie zostanie przyjęta.

6. Sprawozdania o błędach urządzeń

Rysunek ilustruje typowe sprawozdanie dla błędu urządzenia. Rysunek ten służy także jako przykład sprawozdania o wyczerpaniu się czasu oczekiwania na przerwanie, jako że format sprawozdania jest identyczny dla obydwu rodzajów błędów. Poniższy opis sprawozdania o błędzie pamięci dyskowej odwołuje się do poszczególnych wierszy tekstu według ich numeru.

```
1 ***** ENTRY 2. *****
2 DEVICE HARDWARE ERROR
3 LOGGED 30-AUG-83 08:52:52
  *****
4 UNIT IDENTIFICATION
5     UNIT LOGICAL NAME      DK5
6     UNIT PHYSICAL NAME     DK5 (CONTROLLER-0
                             UNIT-5)
7     DEVICE TYPE            SMRK5/SMRK5F/SMRK-3
                             (DISK)
8 SOFTWARE STATUS
9     TASK NAME              ...MOU
10    TASK UIC                1,54
11    TASK START ADDRESS     230600
12    TASK BUFFER ADDRESS    236750
13    TRANSFER SIZE (BYTES)  1000    512.
15    IO FUNCTION ISSUED     READ
16 DEVIVE REGISTERS
17    RKDS    004400
18    RKER    000002 CHECKSUM ERROR
19    RKCS    100744 READ
20    RKWC    177400
21    RKBA    036750
22    RKDA    120001
23 MEDIA ADDRESS AT IO START AT ERROR
24    CYLINDER      0      0
25    TRACK         0      0
```

26	SECTOR	1	1
27	LOGICAL BLOCK	1.	1.
28	ERROR DIAGNOSIS		
29	NOT RECOVERED AFTER 8. RETRIES		
30	VECTORS WITH ACTIVE IO		
31	220		

Rys.2. Raport o błędach urządzeń

Pierwsze trzy wiersze tekstu, oddzielone dwoma wierszami gwiazdek podają informację tytułową:

- Wiersz pierwszy podaje numer błędu
- Wiersz drugi opisuje typ błędu (błąd urządzenia)
- Wiersz trzeci podaje datę i czas zarejestrowania błędu

Następne wiersze danych (UNIT IDENTIFICATION) opisują urządzenie, które spowodowało błąd:

- Wiersz piąty podaje symbol urządzenia
- Wiersz szósty podaje oznaczenie mnemoniczne urządzenia, numer kontrolera oraz fizyczny numer jednostki. Numery jednostek dla jednego kontrolera mogą zmieniać się w granicach od 0 do 7.
- Wiersz siódmy podaje typ urządzenia SMRK5/SMRK5F (DISK).

Następny blok danych (SOFTWARE STATUS) opisuje zadanie, które zainicjowało błędną operację wejścia/wyjścia:

- Wiersz 9 podaje 6-znakową nazwę zadania, które zainicjowało operację wejścia/wyjścia
- Wiersz 10 podaje kod identyfikacyjny użytkownika UIC
- Wiersz 11 i 12 opisują adres startowy zadania i adres buforu wejścia/wyjścia
- Wiersz 13 podaje liczbę bajtów do przesłania
- Wiersz 15 podaje zleconą funkcję wejścia/wyjścia.

Kolejny blok danych - wiersze od 16 do 22 opisują zawartość rejestrów urządzenia w chwili przerwania. Pierwsza kolumna podaje oznaczenie rejestrów urządzenia, w drugiej kolumnie ich

zawartość, w trzeciej zaś, obok rejestru kontroli i stanu CSR oraz rejestru błędu RKER podawana jest funkcja wejścia/wyjścia i rodzaj błędu.

Następny blok raportu (MEDIA ADDRESS - adres na nośniku) ukazuje fizyczny adres nośnika w chwili wykrycia błędu. Linia 27 (LOGICAL BLOCK) podaje numery bloków logicznych LBN w chwili inicjalizacji operacji.

Następny blok raportu (ERROR DIAGNOSIS - linie 28 i 29) ukazuje, czy operacja zakończyła się pomyślnie oraz po ilu powtórzeniach.

Ostatni blok tekstu (linie 30 i 31) zawierają wektory przerwań urządzeń, które wykonywały operacje wejścia/wyjścia. Blok ten nie jest drukowany, jeśli w chwili wykrycia błędów nie były zainicjowane operacje wejścia/wyjścia.

7. Sprawozdanie zbiorcze

Rysunek zawiera przykładowe sprawozdanie zbiorcze. Tego rodzaju sprawozdanie zawiera sumaryczne informacje dotyczące wszystkich błędów zarejestrowanych w systemie (niekoniecznie o wszystkich błędach zawartych w zbiorze wejściowym SYE).

```
#####  
1  SYSTEM ERROR REPORT SUMMARY TOTALS  
#####  
2  DEVICE ERRORS                               40.  
3  MEMORY PARITY ERRORS                       0.  
4  CACHE PARITY ERRORS                        0.  
5  INTERRUPT TIMEOUT ERRORS                   0.  
6  UNEXPECTED TRAPS OR INTERRUPTS            0.  
7  POWER FAILS                                0.  
8  ENTRIES MISSING (SATURATION)              0.  
9      COMMAND LINE USER                      Q=DK1:/FUL  
10     INPUT FILE                             DK1:[1,6]  
                                           ERROR.SYS;1  
11     OUTPUT FILE                            SYO:[6,10]Q;2
```

```
12 DATE OF FIRST ENTRY      30-AUG-83 08:52:19
13 DATE OF LAST ENTRY      30-AUG-83 16:42:52
14 ENTRIES PROCESSED       28.
15 UNRECOGNIZABLE ENTRIES 0.
16XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
17 DETAILED DEVICE ERROR REPORT SUMMARIES
18XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
19 UNIT LOGICAL NAME      DK5
20 UNIT PHYSICAL NAME     DK5 CONTROLLER-0 UNIT-5
21 DEVICE TYPE            SMRK5/SMRK5F/SMRK3
22
23                        [HARD] [SOFT]
24 MOTION/SEEK            0.      1.
25 DATA LATE             10.     0.
26 NOT CATEGORIZED       0.      29.
```

Rys.3. Sumaryczny raport o błędach

Linie 2 do 9 podają liczby zarejestrowanych błędów:

- linia 2 - liczbę błędów urządzeń
- linia 3 - liczbę błędów parzystości pamięci
- linia 4 - liczby błędów parzystości pamięci buforowej (CACHE)
- linia 5 - liczbę błędów wyczerpania limitu czasu
- linia 6 - liczbę zarejestrowanych przerw nie wykorzystywanych w normalnej pracy systemu.
- linia 7 - liczba zaników napięcia zasilającego.
- linia 8 - liczbę błędów niezarejestrowanych, tzn. błędów, jaki powstały podczas zbierania informacji o innych błędach w systemie.
- linia 10 - linia komend SYE użyta do generowania raportu
- linia 11 opisuje zbiór wejściowy SYE, linia 12 zbiór wyjściowy
- linia 12 i 13 podają daty i czas pierwszego i ostatniego uruchomienia zadania ERRLOG

- linia 14 podaje liczbę bloków danych formatowanych przez SYE
- linia 15 podaje liczbę napotkanych błędów, których SYE nie jest w stanie rozpoznać.

Sytuacja taka może zdarzyć się w następujących przypadkach:

1. uruchomiono starą wersję SYE
2. SYE nie obsługuje tego rodzaju błędu urządzenia
3. zbiór wejściowy SYE ma zły format, np. użyto złej wersji PSE

Linia 17 rozpoczyna szczegółowy raport o błędach urządzeń.

Możliwe są następujące typy błędów:

NAGŁÓWEK

OPIS

1. CHECKSUM OR CRC Kontroler wykrył błąd parzystości danych, lub błąd sumy kontrolnej lub błąd CRC.
2. CONTROLLER PARITY Kontroler wykrył błąd parzystości w samym sobie
3. DATA LATE Kontroler wykrył błąd danych spowodowany jedną z dwóch sytuacji:
 - dane były gotowe dla pamięci w czasie, gdy pamięć nie opróżniła jeszcze swego buforu sprzętowego
 - kontroler był gotowy przyjąć dane lecz w buforze pamięci nie było jeszcze danych gotowych do zapisu.
4. DATA PARITY Kontroler wykrył błąd parzystości danych.
5. DEV-TO-MEM PARITY Kontroler wykrył błąd parzystości podczas wykonywa-

- | | |
|------------------------|---|
| | nia zapisu lub zapisu ze sprawdzeniem. |
| 6. DRIVE | Specyficzne błędy jednostki dyskowej np. błąd synchronizacji. |
| 7. DRIVE SELECT | Błędy związane z wyborem urządzenia np. wielokrotny wybór głowic. |
| 8. HEADER | Kontroler wykrył błąd jednostki, który uniemożliwia skuteczne przekazanie danych. Błędami takimi są błędy szukania, błędy porównywania nagłówka, błędy formatu. |
| 9. INTERRUPT TIMEOUT | Zainicjowana operacja nie spowodowała przerwania w wyspecyfikowanym okresie czasu. |
| 10. MOTION/SEEK | Kontroler wykrył mechaniczną niesprawność w jednostce urządzenia powodującą następujące błędy: "szukanie nie zakończone", "błąd szukania" bądź "operacja nie zakończona". |
| 11. NONEXISTENT MEMORY | Kontroler wykrył błąd nieistniejącej pamięci. |
| 12. POWER ERROR | Kontroler wykrył błąd związany z brakiem zasilania. |
| 13. SOFTWARE ERROR | Złe procedury obsługi programowej spowodowały błąd kontrolera. |
| 14. UNIT UNSAFE | Kontroler wykrył błąd, który sprawia, że wybrane urządzenie nie może być użyte. |
| 15. WRITE CHECK | Pojawił się błąd kontroli zapisu. |

16. WRITE CHECK Usiłowano wykonać operacje zapisu na urządzeniu z zablokowanym zapisem.
17. UNCATEGORIZED ERROR Dowolny błąd, który nie należy do żadnej z powyższych kategorii.
- linie 19-25 zawierają opis sumaryczny błędów urządzeń. Podane są nazwy urządzeń i liczby poszczególnych błędów: trwałych (HARD) i korygowalnych (SOFT).

PROGRAM MANIPULACJI ZBIORAMI

Program manipulacji zbiorami (PIP) jest programem użytkowym, działającym na zbiorach w standardzie podsystemu zbiorów systemu DOC PB.

Umożliwia on przesyłanie zbiorów danych między standardowymi urządzeniami DOC PB, a ponadto wykonuje także funkcje sterujące. Niektóre z możliwości dostarczonych przez PIP to:

- kopiowanie zbiorów z jednego urządzenia na inne
- usuwanie zbiorów
- przemianowanie (zmiana nazwy) zbioru
- drukowanie zawartości katalogu
- określanie domniemanego urządzenia i kodu identyfikacji użytkownika, dla dalszych operacji programu manipulacji zbiorami
- odblokowanie zbiorów
- dołączanie wydruków zawartości zbiorów do kolejki PRT&QMG

Działanie programu PIP inicjuje się używając jednej z metod inicjowania programów użytkowych. Wykonanie funkcji programu PIP wywołuje się za pomocą kluczy i podkluczy.

1. Wiersz komendy PIP

Z wyjątkiem numeru wersji, PIP generalnie używa jako domniemania ostatnią wartość, jaka wystąpiła w wierszu komendy. To znaczy, że wykorzystuje wartości określające wielkości domniemane i zmienia domniemanie wraz ze zmianą wartości wprowadzonych.

Następujący wiersz komendy określa nowe wartości domniemane dla każdego zbioru wymienionego w komendzie:

```
>PIP T1.MAC;5,T2,.TSK,6/LI
```

Kolejne wartości domniemane

T1.MAC;5

T2.MAC;5

T2.TSK;5

T2.TSK;6

Reguły określania wartości domniemanych dla PIP
podano w tablicy 1.

Tablica 1.

Wartości domniemane dla specyfikacji zbiorów

<u>Element</u>	<u>Wartości domniemania</u>
dev:	dla pierwszej specyfikacji zbioru jest to urządzenie, na którym zamontowano dysk systemowy (SY:) lub domniemane urządzenie, które określono za pomocą klucza /DF. Dla kolejnej specyfikacji zbioru, można jawnie określić nowe urządzenie lub też PIP przyjmuje jako domniemane urządzenie z poprzedniej specyfikacji
[ufd]	dla pierwszej specyfikacji zbioru, bieżący kod identyfikacji użytkownika - UIC, tzn.: UIC pod którym użytkownik podłączył się do systemu, UIC określony za pomocą komendy monitora - SET lub domniemana wartość UIC określona za pomocą klucza /DF. Dla kolejnej specyfikacji zbioru, można jawnie określić nowy UIC lub też PIP przyjmuje jako domniemany UIC z poprzedniej specyfikacji. Poprawna jest także specyfikacja za pomocą symbolu (*).
;ver	wartością domniemaną dla zbiorów wejściowych jest ostatni numer wersji. Wartością domniemaną, dla zbiorów wyjściowych jest następny, zwiększony o 1 numer wersji lub wersja 1, jeżeli zbioru o podanej nazwie nie ma w katalogu wyjściowym. Jedynym wyjątkiem jest komenda usuwania zbiorów, w której trzeba jawnie podać numer wersji lub specyfikację za pomocą symbolu (*). Jawną specyfikacją numeru wersji oznacza specyfikację ;n - gdzie n większe od zera.

Można użyć numeru wersji ;-1 do określenia najstarszej wersji zbioru. Ostatnia wersja może być określona przez numer wersji ;0 lub ;. Można także zastosować specyfikacje za pomocą symbolu (*).

1.1. Klucze i podklucze programu PIP

1.1.1. Klucze

PIP dopuszcza używanie niektórych kluczy bez podania specyfikacji zbioru. Jednak klucz używany w wierszu komendy, musi występować po specyfikacji zbioru lub UDF (jeśli one występują). Klucz można podać raz dla całej listy specyfikacji zbiorów.

Na przykład:

strng1, strng2, strng3/DE

Klucz /DE dotyczy wszystkich ciągów znakowych i program PIP usuwa każdy podany zbiór z jego katalogu zbiorów użytkownika (UFD).

1.1.2. Podklucze

Podklucze stosowane są lokalnie. Dotyczą one jedynie tej specyfikacji zbioru, która je bezpośrednio poprzedza. W przykładzie poniżej, podklucz nowej wersji (/NV) dotyczy zbioru ASDG.MAC (podklucz /NV używany jest łącznie z kluczem zmiany nazwy /RE).

PIP>*.SMP=PRT2.QRT,ASDG.MAC/NV,KG.RAC/RE
PIP zmienia nazwy zbiorów PRT2.QRT i KG.BAC zachowując jednak związane z nimi numery wersji. Nazwa zbioru ASDG.MAC także zostaje zmieniona, a jego numer wersji równy będzie liczbie o 1 większej od najwyższego numeru wersji zbioru ASDG.SMP.

1.2. Konwencja gwiazdki w specyfikacji zbioru

PIP umożliwia zastosowanie konwencji gwiazdki przez użycie znaku (*) w specyfikacji zbioru. Znak gwiazdki (*) w jednym lub więcej polach specyfikacji zbioru oznacza "wszystkie". Na

przykład: wszystkie zbiory, wszystkie rozsze-
rzenia zbiorów lub wszystkie wersje. Użycie
gwiazdki w niektórych przypadkach jest zabro-
nione. W następnych punktach opisano sposób
użycia gwiazdek w specyfikacjach zbiorów wej-
ściowych i wyjściowych.

1.2.1. Symbol (*) w specyfikacji zbioru wyjściowego

Używanie symbolu (*) w specyfikacji zbiorów
wyjściowych jest ograniczone. Symbol (*) nie
może wystąpić w specyfikacjach zbiorów wyjścio-
wych dla następujących funkcji programu PIP:

- kopiowanie pojedynczego zbioru
- konkatencja zbiorów
- dołączenia do istniejącego zbioru
- aktualizacja (przepisywanie) istniejącego
zbioru
- drukowanie katalogu

Jeżeli użyje się (*) w specyfikacji zbioru wyj-
ściowego dla jednej z powyższych funkcji, zna-
czenie wiersza komendy będzie albo niejednozna-
czne, albo zakłada niezakończoną liczbę zbiorów
wyjściowych.

Na przykład:

```
PIP>[200,200]*.*;*=TEST.DAT
```

PIP będzie próbował stworzyć z pojedynczego
zbioru niezakończoną liczbę zbiorów w katalogu
[200,200].

Kiedy tworzy się kopie kilku zbiorów, specyfi-
kacja wyjściowa musi być w postaci *.*;* lub
domniemana.

We wszystkich przypadkach, w których dozwolone
jest użycie gwiazdki w specyfikacji zbioru wyj-
ściowego, określenie za pomocą gwiazdek katalo-
gu UFD w postaci [*,*](nie [n,*], ani [*,*])
oznacza, że UFD wyjściowy jest taki sam jak wej-
ściowy.

1.2.2. Sybol (π) w specyfikacji zbioru wejściowego

PIP umożliwia używanie gwiazdek w specyfikacjach zbiorów wejściowych w następujący sposób:

π.π;π	oznacza wszystkie wersje wszystkich zbiorów
π.DAT;π	oznacza wszystkie wersje, wszystkich zbiorów o rozszerzeniu DAT
TEST.π;π	oznacza wszystkie wersje i wszystkie rozszerzenia zbiorów o nazwie TEST
TEST.DAT;π	oznacza wszystkie wersje zbioru TEST.DAT
π.π	oznacza ostatnie wersje wszystkich zbiorów
π.DAT	oznacza ostatnie wersje wszystkich zbiorów o rozszerzeniu .DAT
TEST.π	oznacza ostatnie wersje zbiorów o dowolnym rozszerzeniu i nazwie TEST

PIP umożliwia używanie symbolu (π) w określe-
niu UFD:

[π,π]	oznacza wszystkie kombinacje numerów grup i użytkowników w grupach od 1 do 377 oktalnie
[n1,π]	oznacza wszystkich użytkowników w grupie n1
[π,n2]	oznacza wszystkich użytkowników o numerze n2, ze wszystkich grup

2. Komendy programu PIP

PIP umożliwia kopiowanie zbioru w standardzie DOC PB, dostarczając jednocześnie różnorodne funkcje zarządzania zbiorami.

2.1. Kopiowanie zbiorów systemu DOC PB

W celu skopiowania zbiorów w standardzie DOC PB, należy wprowadzić wiersz komendy bez żadnych kluczy.

Poniżej przedstawiono najprostszy przykład wiersza komendy PIP:

```
outfil=infil
```

Jedynym kluczem, jaki można użyć przy kopiowaniu zbiorów w standardzie DOC PB, jest klucz łączenia zbiorów. Jeśli przy kopiowaniu zostanie jawnie określony zbiór wyjściowy i dwa (lub więcej) zbiory wejściowe bez żadnych kluczy, PIP automatycznie dołącza klucz /ME. Ponieważ klucz /ME jest związany z kopiowaniem, został opisany w tym punkcie osobno (a nie wraz z innymi kluczami).

Ogólna postać wiersza komendy PIP jest następująca:

```
outfil=infil1[,infil2,...,infil][[/ME][subsw]]
```

gdzie:

outfil	specyfikacja zbioru wyjściowego
infil	specyfikacja zbioru wejściowego
/ME	specyfikacja klucza łączenia zbiorów
/subsw	specyfikacja dowolnego z podkluczy jakie można wprowadzić jako część podstawowego wiersza komendy lub razem z kluczem /ME. Podklucze opisano w tabelicy 2. Podklucze mogą być użyte zarówno w specyfikacji zbioru wejściowego jak i wyjściowego. Jeżeli podklucz zostanie użyty w specyfikacji zbioru wejściowego, dotyczy on tylko tego zbioru. Jeżeli podklucz zostanie użyty w specyfikacji zbioru wyjściowego, dotyczy całej listy specyfikacji wejściowych.

Przykłady:

1. PIP>DK1:SAMP.DAT=DK2:TEST.DAT
Kopiowanie ostatniej wersji zbioru TEST.DAT (w bieżącym UFD) z DK2: na DK1: jako zbioru SAMP.DAT
2. PIP>DK1:[*,*]=DKO:[11,*]
Kopiowanie wszystkich zbiorów wszystkich użytkowników z grupy o numerze 11 z DKO: na

DK1.: Zbiory zostają zapisane w takim samym UFD na DK1: w jakim znajdowały się na DKO:

3. PIP>LP:=#.LST
Kopiowanie wszystkich wersji wszystkich zbiorów z rozszerzeniem .LST, z bieżącego UFD na drukarkę wierszową. Jeśli w systemie zainstalowane jest zadanie dołączania wydruków do kolejki PRT&QMG, zamiast tej komendy można użyć klucz /SP. Wiersz komendy wykorzystujący /SP ma następującą postać:
PIP>#.LST/SP
4. PIP>DK1:SAMP.DAT=DK2:TEST.DAT;1,NEW.DAT;2/ME
Łączenie wersji 1 zbioru TEST.DAT i wersji 2 zbioru NEW.DAT z urządzenia DK2:, tworzenie zbioru SAMP.DAT na DK1: w bieżącym katalogu.
5. PIP>DK1:=SY:TESTPROG.MAC,OBJ
Kopiowanie ostatnich wersji TESTPROG.MAC i TESTPROG.OBJ z urządzenia systemowego (SY:) na DK1: (zarówno dla zbiorów wejściowych i wyjściowych katalog bieżący, przyjęto jako domniemany)
6. PIP>DK1:=DKO:#.DAT;#
Kopiowanie wszystkich wersji, wszystkich zbiorów o rozszerzeniu .DAT w bieżącym katalogu z DKO: na DK1:
7. PIP>DKO:=[200,10]#.#:#
Kopiowanie wszystkich zbiorów z katalogu [200,10], z domniemanego urządzenia na urządzenie DKO:, do bieżącego katalogu.
8. PIP>DPO:[200,10]=DKO:#.#:#
Kopiowanie wszystkich zbiorów z DKO:, z bieżącego katalogu, na DPO:[200,10]. Użytkownik musi mieć prawo zapisu w katalogu [200,10]

Tablica 2

Komendy kopiowania i podklucze łączenia zbiorów

<u>Podklucz</u>	<u>Opis</u>
/BL:n[.]	przydział bloków - określa liczbę(n) bloków w ciągłym obszarze, przydzielanych: wstępnie zbiorowi wyjściowemu. Jako (n) można podać zarówno wartość oktalną jak i dziesiętną na końcu. Podklucz /BL:n używa się przy kopiowaniu ciągłych zbiorów zmieniając ich długość
/CO	ciągły zbiór wyjściowy - informuje, że zbiór wyjściowy ma być ciągły. Podczas kopiowania ciągłych zbiorów z taśmy magnetycznej (na przykład obrazów zadania), należy podać zarówno /CO jak i /BL:n. Podklucz /BL:n umożliwi programowi PIP określenie długości zbioru wejściowego podczas przydzielania miejsca na zbiór (PIP przydziela miejsce na zbiór przed wykonaniem operacji kopiowania). Długość zbiorów na taśmie magnetycznej jest zapisana w etykiecie kończącej taśmę.
/-CO	nieciągły zbiór wyjściowy - informuje, że zbiór wyjściowy nie musi być ciągły. Jeżeli nie poda się żadnego z powyższych kluczy, PIP przez domniemanie przyjmuje rozmiar i atrybuty zbioru wejściowego.
/FO	kod identyfikacji właściciela zbioru wyjściowego - informuje, że zbiór wyjściowy będzie miał taki sam kod identyfikacji właściciela, jak numer katalogu użytkownika (UFD), do któ-

regu zapisany będzie zbiór wyjściowy. Jeżeli nie podamy klucza /FO, kod identyfikacji właściciela dla wszystkich nowych zbiorów będzie taki sam, jak bieżący kod identyfikacji użytkownika, z którym wykonuje się program PIP niezależnie od katalogu, w którym zostaną zapisane te zbiory. Poklucz /FO można używać zarówno w komendach kopiowania jak i łączenia.

Przykłady:

1. Jeżeli bieżącym kodem identyfikacji użytkownika jest [1,1] to komenda:
DKO:[200,200]=DK1:[200,200]TEST.DAT
tworzy nowy zbiór w katalogu [200,200] na DKO:. Zbiór ten ma kod identyfikacji [1,1].

Natomiast komenda:

DKO:[200,200]=DK1:[200,220]TEST/FO
tworzy zbiór z kodem identyfikacji właściciela [200,200]. Kiedy używa się klucz /FO, bieżący kod identyfikacji użytkownika, musi dawać prawo zapisu do wszystkich katalogów wyjściowych

2. DK1:[*,*]/FO=DPO:[13,10],[32,10],
[34,10]

Kopiowanie wszystkich zbiorów z podanych katalogów wejściowych, do odpowiednich katalogów na DK1:. Kody identyfikacji właściciela dla zbiorów wyjściowych są takie same jak katalogi wyjściowe

3. DK1:[*,*]=DKO:[*,10]*.MAC/FO
Kopiowanie wszystkich zbiorów o rozszerzeniu .MAC, o numerach użytkowników równych 10, ze wszystkich grup na dysku DK1:, zachowując UFD. Kod identyfikacji właściciela, dla każdego zbioru jest taki, jak UFD zbioru.

/SU zastępowanie zbioru - pozwala skopiować jeden lub więcej zbiorów wejściowych do katalogu wyjściowego, jeśli zbiór o tej samej nazwie, rozszerzeniu i wersji już istnieje. Istniejący zbiór jest usuwany, a nowy zbiór tworzony jest w jego miejsce z danymi ze zbioru wejściowego (zbiorów wejściowych). Nazwa, rozszerzenie i numer wersji zbioru wyjściowego pozostaje nie zmieniona, zmienia się natomiast identyfikator zbioru. Mogą także ulec zmianie atrybuty zbioru wyjściowego, które brane są z pierwszego zbioru wejściowego, a także liczba bloków przydzielonych do zbioru wyjściowego (może być ich więcej lub mniej niż posiadał istniejący zbiór).

/NV nowa wersja - powoduje, że numer wersji skopiowanego zbioru wyjściowego będzie o 1 większy, od ostatniej wersji zbiorów już istniejących w katalogu wyjściowym. Jeżeli takiego zbioru nie było w katalogu, otrzymuje on numer wersji 1. Rysunek 1 przedstawia sposób używania podklucza /NV (podklucza /NV nie trzeba podawać, jeśli katalogi zbiorów wejściowych i wyjściowych są takie same)

2.2. Funkcje sterujące programu PIP

PIP posiada kilka kluczy i podkluczy do sterowania przetwarzaniem zbiorów. Dzięki nim, użytkownik korzystający z programu PIP może usuwać zbiory, drukować na terminalu zawartość katalogów i określać prawa dostępu do zbiorów.

W wierszu komendy nie zawierającej specyfikacji zbioru można użyć dwa klucze: /ID i /LI (tzn.: że mogą one być podane samodzielnie).

W jednym wierszu komendy można użyć tylko jeden klucz. Można jednak zastosować więcej niż jeden podklucz.

Liczby użyte z kluczami i podkluczami, przyjęte są przez domniemanie jako oktalne. Jeżeli użytkownik chce stosować liczby dziesiętne, musi na końcu liczby dodać kropkę dziesiętną. Tablica 3 zawiera spis kluczy i podkluczy oraz ogólny opis funkcji przez nie wykonywanych. Podklucze opisano wraz ze związanymi z nimi kluczami (dokładny opis kluczy i podkluczy znajduje się w oddzielnych podpunktach).

Katalogi przed skopiowaniem

Katalog wejściowy [201,201]	Katalog wyjściowy [100,100]
RICK.DAT;1	RICK.DAT;2 RICK.DAT;4

Katalogi po skopiowaniu bez klucza /NV
(zachowany numer wersji)

Katalog wejściowy [201,201]	Katalog wyjściowy [100,100]
RICK.DAT;1	RICK.DAT;2 RICK.DAT;4 RICK;1

Została użyta komenda:

DK1:[100,100]=[201,201]RICK.DAT;1

Katalogi po skopiowaniu z kluczem /NV

Katalog wejściowy [201,201]	Katalog wyjściowy [100,100]
RICK.DAT;1	RICK.DAT;2 RICK.DAT;4 RICK.DAT;5

Została użyta komenda:

DK1:[100,100]=DK1:[201,201]RICK.DAT;1/NV

UWAGA

Numer wersji podawany razem z podkluczem /NV musi być wyspecyfikowany w sposób jawny lub domniemany (nie może być użyty symbol (*))

Rys.1. Zastosowanie komendy kopiowania oraz klucza /NV

Tablica 3

Klucze i podklucze programu PIP

<u>Klucz</u>	<u>Podklucz</u>	<u>Funkcja</u>
/AP		dopisywanie zbioru /zbiorów/ na koniec istniejącego zbioru
	/FO	określenie kodu identyfikacji właściciela zbioru
/BR		drukowanie katalogu zbiorów w formacie skróconym (opcjonalny sposób drukowania za pomocą klucza /LI)
/BS:n[.]		określa długość bloku na taśmie magnetycznej
/CD		zapisanie daty powstania zbioru wyjściowego równej dacie powstania zbioru wejściowego. Gdy klucz nie zostanie podany zapisze się data dokonania zapisu do zbioru wyjściowego
/DE		usuwanie jednego lub więcej zbiorów
	/LD	drukowanie specyfikacji usuniętych zbiorów
/DF		zmiana domniemanego urzędnika albo katalogu
/EN		włączenie synonimu zbioru do katalogu
	/NV	numer wersji zbioru będzie o 1 większy od ostatniej wersji
/EOF[:blk:byt]		specyfikacja znacznika końca zbioru
/FI:flnum:sgnum		dostęp do zbioru przez jego identyfikator
/FR		drukowanie rozmiaru wolnej przestrzeni na określonym tomie i najdłuższego wolnego obszaru ciągłego na tym tomie

<u>Klucz</u>	<u>Podklucz</u>	<u>Funkcja</u>
/FU[:n[.]]		drukowanie katalogu w pełnym formacie (opcjonalny sposób drukowania za pomocą klucza /LI)
/ID		drukowanie numeru wersji programu PIP
/LI		drukowanie katalogu zbiorów
/ME		łączenie w jeden zbiór dwu lub więcej zbiorów
	/BL:n[.]	przydzielenie n ciągłych bloków
	/CO	zbiór /zbiory/ wyjściowy ma być ciągły
	/FO	określenie kodu identyfikacji właściciela zbioru wejściowego
	/NV	numer wersji zbioru będzie o 1 większy od ostatniego numeru wersji
	/SU	zastąpienie istniejącego zbioru
/MM		zablokowanie komunikatów programu PIP, o niektórych błędach
/PR		zmiana kodów ochrony zbioru
	/FO	określenie kodu identyfikacji właściciela zbioru
	/GR[:RWED]	ustawienie kodów ochrony zbioru na poziomie grupy użytkowników /prawo do: czytania, zapisu, rozszerzenia, usunięcia zbioru/
	/OW[:RWED]	ustawienie kodów ochrony zbioru na poziomie właściciela (prawo do: czytania, zapisu, rozszerzenia, usunięcia zbioru)

<u>Klucz</u>	<u>Podklucz</u>	<u>Funkcja</u>
/PU[:n[.]]		usunięcie wszystkich (poza ostatnią) wersji danego zbioru
	/LD	drukowanie specyfikacji usuwania zbiorów
/RE		zmiana nazwy, rozszerzenia albo wersji zbioru
/RM		usuwanie pozycji z katalogu
/RW		przewinięcie taśmy magnetycznej
/SB		kopiowanie rekordów z taśmy magnetycznej na dysk, z przekroczeniem granic bloków
/SD		selektywne usuwanie zbiorów. Użytkownik przed usunięciem zbioru jest pytany, czy dany zbiór ma być usunięty
/SP[:n[.]]		ustawianie wydruków zbioru /zbiorów/ do kolejki PRT&QMG
/SR		pozwala na dzielony dostęp w celu czytania do zbioru, który jest otwarty dla zapisu przez innego użytkownika lub zadanie
/TB		drukowanie całkowitej liczby bloków zajmowanych przez zbiory w katalogu, razem z całkowitą liczbą bloków przydzielonych oraz z liczbą zbiorów w tym katalogu (opcjonalny sposób drukowania za pomocą klucza /LI)
/TR		obcięcie zbioru /zbiorów/ do logicznego końca zbioru
/UF		tworzenie katalogu zbiorów użytkownika (UPD) na tomie,

<u>Klucz</u>	<u>Podklucz</u>	<u>Funkcja</u>
/UN /UP		na który zbiór jest przepisywany odblokowanie zbioru aktualizacja /przepisanie/ istniejącego zbioru
	/FO	określenie kodu identyfikacji właściciela zbioru

PROGRAM FORMATOWANIA DYSKÓW

Program formatowania dysków (dyskowych nośników informacji - FMT) formatuje i sprawdza formatowanie dysków elastycznych, kaset dyskowych oraz dysków pakietowych. Urządzenia na których wykonywane jest formatowanie mają następujące nazwy:

DB:, DK:, DM:, DP:, DR: i DY:

Program FMT może być używany w systemie operacyjnym DOC PB pod warunkiem, że w czasie generacji systemu dołączono do egzekutora procedury diagnostyczne.

Nośniki mogą być formatowane w dwóch trybach: automatycznym i manualnym. W trybie automatycznym formatowaniu podlega cały nośnik, a w trybie manualnym formatowaniu podlegają poszczególne (wskazane przez użytkownika) sektory lub ścieżki (dla dysku DM:). Domniemanym trybem pracy jest tryb automatyczny.

Tryb manualny pozwala w niektórych przypadkach odzyskiwać poszczególne sektory (spośród uszkodzonych).

UWAGA

Używane w tym opisie określenie "sektor", równoważne jest określeniu "blok fizyczny"

Formater wykonuje następujące funkcje:

- zapisuje nagłówki przed każdym sektorem na dysku
- sprawdza zawartość nagłówków sektorów (w trybie weryfikacji formatowania)
- ustala gęstość zapisu dla dysku elastycznego DY:
- formater może testować jakość formatowanego nośnika. Po osiągnięciu zadeklarowanej przez użytkownika maksymalnej /dopuszczalnej/ liczby błędów formatowania jest ono przerywane /po wydruku odpowiedniego komunikatu/, co stanowi pewną formę testu akceptacyjnego
- inicjuje działanie programu testowania nośników dyskowych - BAD /pod warunkiem, że egzekutor ma wygenerowaną opcję kreowania zadań/

1. Rozpoczęcie i zakończenie pracy formatera

Aby zainicjować pracę formatera (FMT) należy /po przynagleniu systemowym/ podać odpowiednią komendę.

Postać komendy inicjującej zadanie FMT jest następująca:

>FMT ddn:[/klucz1.../klucz2]

gdzie: dd: jest nazwą dysku, a n jest numerem jednostki dyskowej

Poniżej podane zostały klucze interpretowane przez formater.

/BAD inicjacja zadania BAD (programu testowania nośników dyskowych) - pod warunkiem, że jest ono zainstalowane w systemie.

Należy zaznaczyć, że realizacja tego klucza zależy od tego, czy system operacyjny /wygenerowana jego wersja/ pozwala na inicjację zadania przez inne zadanie /zadania potomne/

/DEND wybiera podwójną lub pojedynczą gęstość zapisu na dysku elastycznym SMRX2

/ERL określa maksymalną ilość błędów dopuszczalnych w trakcie formatowania nośnika

/MAN ustala manualny tryb pracy i umożliwia formatowanie sektorów /ścieżek/ wskazywanych przez użytkownika

/OVR ustala, że zbiór opisujący fabrycznie wykryte sektory uszkodzone (MDBSF) należy zignorować. Zbiór ten zapisywany jest jedynie na dyskach DM: i DR: /na ostatnich ścieżkach/, a zatem klucz ten dotyczy tylko tych błędów

/WLT pozwala zmieniać zbiór opisujący fabrycznie wykryte sektory uszkodzone (MDBSF), aby dopisać informacje o uszkodzonych sektorach wykrytych w trakcie pracy formatera

/VE wskazuje, że formater ma sprawdzić /po zaformatowaniu nośnika/ czy nagłówki sektorów są poprawnie zapisane

/@Y wskazuje formaterowi, że jego praca sterowana jest ze zbioru komend. W tym przypadku FMT nie pozwala na interwencje operatora w trakcie swej pracy.

2. Tryby pracy formatera

Formater może pracować w dwóch trybach:

- w trybie automatycznym, w którym formatowaniu podlega cały nośnik (sektor po sektorze) oraz
- w trybie manualnym, w którym formatowane są poszczególne, wskazane przez użytkownika sektory (lub ścieżki - dysk DM:)

Jeśli nie podano klucza /MAN (inicjującego pracę w trybie manualnym), to FMT pracuje w trybie automatycznym.

Jeżeli w trakcie zapisywania nagłówka sektora wystąpi błąd, to FMT powtarza ponownie operacje. Jeżeli powtórny zapis również kończy się błędem, to FMT oznacza ten sektor jako uszkodzony i drukuje poniższy komunikat:

ERROR WRITING HEADER

Jeżeli użyto klucza /VE i w trakcie sprawdzania wystąpił błąd, to wypisany zostanie następujący komunikat:

ERROR READING HEADER

lub

HEADER COMPARE ERROR

a potem FMT kontynuuje sprawdzanie nagłówków.

2.1. Tryb automatyczny

Jeżeli formater zainicjowany został do pracy w trybie automatycznym /nie użyto klucza /MAN/, to drukuje następujący komunikat:

WARNING - DATA WILL BE LOST ON ddn: ##
CONTINUE [Y OR N]

co po polsku znaczy:

UWAGA - DANE NA DYSKU ddn: BĘDĄ UTRACONE ##
CZY KONTYNUOWAC [TAK LUB NIE]

Jeśli użytkownik poda odpowiedź twierdzącą - Y (YES = TAK), to FMT drukuje komunikat:

START FORMATTING

i zaczyna formatować nośnik oraz realizuje klu-
cze podane w wierszu komendy. Po odpowiedzi
przeczącej - N (NO = NIE) lub odpowiedzi pustej
/jedynie znak karetki <CR>, FMT przerywa pracę
i sterowanie wraca do minitora systemowego.
Praca w trybie automatycznym zależy od typu u-
rządzenia na którym formatowany jest nośnik.
Pracę w tym trybie na różnych urządzeniach o-
pisuje punkt 3.

2.2. Tryb manualny

Jeśli wybrany został manualny tryb pracy /po-
dano klucz /MAN/, to formater drukuje komunikat:

WARNING - DATA WILL BE LOST ON ddn:

CONTINUE [Y OR N]

co po polsku znaczy:

UWAGA - DANE NA DYSKU ddn: BĘDĄ UTRACONE

CZY KONTYNUOWAC [TAK LUB NIE]

Jeśli użytkownik poda odpowiedź twierdzącą - Y
(YES = tak), to FMT drukuje komunikat:

ENTERING MANUAL MODE

a następnie drukuje następujące przynaglenie:

CYLINDER=

TRACK =

SECTOR =

co po polsku znaczy:

CYLINDER=

ŚCIEŻKA =

SEKTOR =

Po wprowadzeniu odpowiedzi FMT formatuje okre-
ślony sektor lub ścieżkę. Jeśli wiodącym zna-
kiem odpowiedzi nie jest wskaźnik liczby oktal-
nej - (#), to FMT przyjmuje, że podano liczbę
dziesiętną.

Jeżeli wprowadzony parametr nie mieści się w
odpowiednim dla danego urządzenia przedziale,
to FMT drukuje komunikat o błędzie i przerywa
wykonywanie aktualnej komendy.

FMT realizuje manualny tryb pracy jednakowo na wszystkich typach dysków z wyjątkiem dysków DM: (SMRK6 i SMRK7), na których formatowane są wyspecyfikowane ścieżki a nie sektory.

Na przykład:

FMT>DMO:/MAN

ta komenda wywołuje przynaglenia:

WARNING - DATA WILL BE LOST ON DMO: ##
CONTINUE [Y OR N]

ENTERING MANUAL MODE

CYLINDER=

TRACK =

Po czym FMT formatuje podaną ścieżkę. Zakresy dozwolonych wartości parametrów:

SMRP2/SMRPR2 0-9 0-19 0-199

SMRK5/SMRK5F 0-11 0-1 9-199

3. Urządzenia formatowane przez FMT

W tabelicy 1 przedstawione zostały urządzenia formatowane przez program FMT oraz ich skrócone nazwy.

Tablica 1

Urządzenia formatowane programem FMT

Nazwa dysku	Nazwa skrócona
SMRP4 - dysk pakietowy	DB:
SMRP5 - dysk pakietowy	DB:
SMRP6 - dysk pakietowy	DB:
SMRK5 - dysk kasetowy wymienny	DK:
SMRK5F - dysk kasetowy stały	DK:
SMRK6 - dysk kasetowy wymienny	DM:
SMRK7 - dysk kasetowy wymienny	DM:
SMRPR2 - dysk pakietowy	DP:
SMRP2 - dysk pakietowy	DP:
SMRP3 - dysk pakietowy	DP:

SMRM2 - dysk pakietowy DR:
SMRM2 - dysk pakietowy DR:
SMRX2 - dysk elastyczny DY:
Formater wymaga, aby w czasie formatowania
dyski nie były zamontowane (w sensie systemu
operacyjnego DOC PB).

3.1. DK: (dyski SMRK5/SMRK5F)

W czasie formatowania dyski DK: formater zapisuje nagłówki każdego sektora osobno, aż do zaformatowania całego dysku. Jeżeli w czasie zapisu wystąpi błąd, to formater powtarza zapis dwukrotnie przed ewentualnym wydrukowaniem komunikatu na terminalu.

Jeżeli podano klucz weryfikacji /VE, to FMT sprawdza odcinki nośnika o długości 12 sektorów, aż do sprawdzenia całego nośnika. Jeżeli wystąpi błąd, to FMT przystępuje do sprawdzania nagłówków indywidualnie (sektor po sektorze). Po ustaleniu, który sektor ma błędnie zapisany nagłówek, na terminalu wypisywany jest (dla każdego takiego przypadku) komunikat o złym zaformowaniu sektora.

3.2. DP: (dyski SMRPR2/SMRP2/SMRP3)

W czasie formatowania dysku DP: formater zapisuje po 10 nagłówków sektorów jednocześnie, aż do zaformatowania całego nośnika. Jeżeli w trakcie zapisu nagłówków wystąpi błąd, to formater zaczyna zapisywać nagłówki kolejnych sektorów indywidualnie (w ramach 10 nagłówków wśród których wystąpił błąd), aby wykryć sektor w którym ten błąd wystąpił.

Jeżeli podano klucz weryfikacji /VE, to MFT sprawdza odcinki nośnika o długości 10 sektorów, aż do sprawdzenia całego nośnika. Jeżeli wystąpi błąd, to MFT przystępuje do sprawdzenia nagłówków indywidualnie (sektor po sektorze). Po ustaleniu, który sektor ma błęd-

nie zapisany nagłówek, na terminalu wypisywany jest (dla każdego takiego przypadku) komunikat o złym zaformatowaniu sektora.

4. Klucze formatera

W następujących punktach opisane są klucze używane w formaterze. Opis kluczy zawiera informacje o ich funkcjach, ograniczeniach występujących przy formatowaniu niektórych urządzeń oraz domniemane wartości kluczy.

4.1. /BAD

Klucz /BAD inicjuje wykonanie programu testowania nośników dyskowych (zadanie BAD). Jeśli zadanie BAD nie jest zainstalowane, to FMT drukuje komunikat na terminalu i przerywa swoją pracę (następuje powrót do monitora systemowego). Komenda FMT zawierająca klucz /BAD ma następujący format:

```
FMT>dev:/BAD
```

4.2. /DENS

Klucz /DENS ustala gęstość zapisu na dysku elastycznym DY: (dopuszczalne są gęstości: pojedyncza - słowa kluczowe LOW lub SINGLE oraz gęstość podwójna - słowa kluczowe HIGH lub DOUBLE). Jako domniemaną przyjmuje się gęstość pojedynczą (LOW). Komenda FMT zawierająca klucz /DENS ma następujący format:

```
FMT>DY:/DENS=HIGH (lub DOUBLE)
```

```
FMT>DY:/DENS=LOW (lub SINGLE)
```

4.3. /ERL

Klucz /ERL ustala maksymalną ilość błędów dopuszczalnych przez formater w trakcie formatowania nośnika. Po osiągnięciu tej liczby formater drukuje komunikat na terminalu i przerywa swoją pracę. Dopuszcza się liczby z przedziału 1 - 256. Jako domniemaną przyjmuje się liczbę 256.

Komenda FMT zawierająca klucz /ERL ma następujący format:

```
FMT)dev:/ERL=n
```

4.4. /MAN

Klucz /MAN przełącza formater na manualny tryb formatowania. Pozwala to formatować poszczególne sektory na nośniku (dla dysku DM: poszczególne ścieżki). Jeśli wiodącym znakiem prowadzonej liczby nie jest znak (#), to formater zakłada, że wprowadzana liczba jest liczbą dziesiętną.

W trybie manualnym FMT wypisuje następujące przynaglenie:

```
** WARNING - DATA WILL BE LOST ON DK1: **  
CONTINUE [Y OR N]  
ENTERING MANUAL MODE  
CYLINDER=  
TRACK   =  
SECTOR  =
```

```
OPERATION COMPLETE
```

Komenda FMT zawierająca klucz /MAN ma następujący format:

```
FMT)dev:/MAN
```

4.5. /OVR

Klucz /OVR pozwala formaterowi ignorować zbiór opisujący fabrycznie wykryte sektory uszkodzone (MBSF). W czasie formatowania dysków DM: i DR: formater zapisuje wskaźnik uszkodzenia sektora w jego nagłówku, jeżeli jest on tak zaznaczony w zbiorze MBSF. W czasie sprawdzania formatowania na terminalu drukowany jest komunikat informujący o tym, że w zbiorze MBSF sektor oznaczony jest jako uszkodzony. Klucz /OVR pozwala zignorować zbiór MBSF (w czasie weryfikacji). Klucz ten dotyczy tylko dysków DM: i DR:.

Komenda FMT zawierająca klucz /OVR ma następujący format:

```
FMT)dev:/OVR
```

4.6. /VE

Klucz /VE pozwala upewnić się, czy formatowanie zostało prawidłowo wykonane. Formater czyta w tym celu zapisane przez siebie nagłówki i sprawdza, czy informacja zapisana w nich jest zgodna z oczekiwaną.

Komenda FMT zawierająca klucz /VE ma następujący format:

```
FMT>dev:/VE
```

4.7. /WLT

Klucz /WLT użyty wraz z kluczem /VE dla dysków DM: i DR:, pozwala użytkownikowi wskazać, że dozwolony jest ponowny zapis zbioru opisującego fabrycznie wykryte sektory uszkodzone (MDBSF). Informacja o uszkodzeniu sektora zapisywana jest również w nagłówkach uszkodzonych sektorów.

Klucz /WLT wymaga podania liczby dziesiętnej (oznaczonej niżej jako n), która stanowi numer porządkowy dysku pakietowego.

Komenda FMT zawierająca klucz /WLT ma następujący format:

```
FMT>dev:/WLT:n
```

4.8. /@Y

Klucz /@Y wskazuje formaterowi, że komendy pobierane są ze zbioru komend. W tym przypadku formater nie drukuje na terminalu żadnych komunikatów (aż do zakończenia operacji) a ponadto nie zezwala na interwencje operatora.

Komenda FMT zawierająca klucz /@Y ma następujący format:

```
FMT>dev:/@Y
```


POSTAWOWE KOMENDY EDYTORA EDI

Komendy podstawowe EDI wymienione w tablicy 1 pozwalają na tworzenie zbioru, modyfikację zbioru poprzez dodawanie, usuwanie lub wymianę jego zawartości oraz umożliwiają zakończenie edycji po wykonaniu żądanych operacji.

Tablica 1.

Podstawowe komendy EDI.

Komenda	Format komendy	Opis
ADD	Add tekst	Dodaje tekst na koniec bieżącego wiersza.
ADD & PRINT	AP tekst	Dodaje tekst na koniec wiersza bieżącego i drukuje wiersz wynikowy.
BOTTOM	Bottom	Przesuwa wskaźnik wiersza bieżącego na koniec bieżącego bloku (w trybie blokowym) lub na koniec zbioru (w trybie wierszowym).
CHANGE	[n]Change/tekst1 /tekst2[/]	Zamiana tekst1 na tekst2 n-razy w wierszu bieżącym.
CTRL/Z	^Z	Zamyka zbiory i kończy sesje edycji.
DELETE	Delete[n] lub Delete [-n]	Usuwa wiersz bieżący i n-1 wierszy po nim następujących, jeśli n jest znakiem (+); usuwa n wierszy poprzedzających wiersz bieżący, jeśli n jest ze znakiem (-). [-n] działa tylko w trybie blokowym.

DELETE & PRINT	DP[n]lub DP[-n]	Działa tak samo jak DELETE, poza tym drukuje nowy wiersz bieżący.
<ESCape >	Znak ESC	Zamyka poprzedni wiersz, który staje się wierszem bieżącym /tylko w trybie blokowym/. Działa tak jak NP-1.
EXIT	Exit [spec.-zbioru]	Zamyka zbiory, zmieniając nazwę zbioru wyjściowego i kończy sesję edycji.
INSERT	Insert [tekst]	Ustawia tekst bezpośrednio za wierszem bieżącym. Jeśli tekst nie jest wyspecyfikowany, EDI wchodzi w tryb wprowadzania.
LOCATE	[n]Locate tekst	Wyszukuje n-te pojawienie się tekstu. W trybie blokowym przeszukiwanie zatrzymuje się na końcu bieżącego bloku.
NEXT	Next[n]lub Next [-n]	Ustanawia nowy wiersz bieżący w odległości n linii od bieżącego wiersza.
NEXT & PRINT	NP[n]lub NP[-n]	Ustawia i drukuje nowy wiersz bieżący.

PRINT	Print[n]	Drukuje wiersz bieżący i n-1 następných wierszy. Ostatni wypisany wiersz jest nowym wierszem bieżącym.
RENEW	RENew[n]	Zapisuje bieżący blok do zbioru wyjściowego i wczytuje nowy blok ze zbioru wejściowego /tylko w trybie blokowym/.
<CR>	powrót karetki	Drukuje następny wiersz, który staje się nowym wierszem bieżącym; powoduje wyjście z trybu wprowadzania. Działa tak jak NP+1.
RETYPE	Retype tekst	Zamienia cały wiersz bieżący na tekst; lub usuwa wiersz bieżący, jeśli tekst nie został wyspecyfikowany.
TOP	Top	Przesuwa wskaźnik wiersza bieżącego na początek bieżącego bloku /w trybie blokowym/ lub na początek zbioru /w trybie wierszowym/. W trybie wierszowym TOP tworzy nową wersję zbioru za każdym razem, gdy komenda ta jest wywołana.
TOP OF FILE	TOF	Powraca na początek zbioru wejściowego i zachowuje wszystkie strony poprzednio redagowane. TOF tworzy nową wersję zbioru za każdym wywołaniem komendy. TOF wczytuje nowy blok do

bufora po zapisaniu poprzedniego bloku do zbioru wyjściowego.

1. Komendy sterujące

Komendy sterujące pozwalają na odblokowanie lub zablokowanie pewnych szczególnych funkcji EDI. Pośród tych funkcji znajdują się uprzednio wymienione tryby dostępu do tekstu: blokowy i wierszowy oraz opcja automatycznego wyprzedzania przy komendach typu lokalizującego. Komendy sterujące wymienione są w tabelicy 2.

Tablica 2.

Komendy sterujące edytora wierszowego EDI.

Komenda	Format	Opis
BLOCK ON/OFF	Block [ON]lub Block OFF	Klucz wybierający tryb dostępu do tekstu.
CONCATENATION CHARACTER	CC [znak]	Zmienia znak wyspecyfikowany w wierszu komendy /domniemanym znakiem jest &/.
OPEN SECONDARY	OPens spec.-zbioru	Otwiera wyspecyfikowany pomocniczy zbiór wejściowy.
OUTPUT ON/OFF	OUtput ON lub OUtput OFF	Kontynuuje lub przerywa przesyłanie tekstu do zbioru wyjściowego /w trybie wierszowym/.

SELECT PRIMARY	SP	Ustawia zbiór główny jako zbiór wejściowy.
SELECT SECONDARY	SS	Wybiera otwarty zbiór pomocniczy jako zbiór wejściowy.
SIZE	SIZE n	Określa maksymalną liczbę wierszy jaką można będzie wczytać do bufora blokowego.
TAB	Tab [ON] lub Tab OFF	Przełączanie automatycznej tabulacji na włączoną lub wyłączoną.
UPPER CASE ON/OFF	UC [ON] lub UC OFF	Zezwolenie lub zablokowanie konwersji znaków dolnego rejestru wprowadzonych z terminala na znaki górnego rejestru.
VERIFY ON/OFF	Verify [ON] lub Verify OFF	Wybiera opcje sprawdzania efektów działania komend lokalizujących i modyfikujących.

2. Komendy lokalizujące (sterujące wskaźnikiem wiersza)

Podczas operacji redagowania EDI posiada wskaźnik, który identyfikuje wskaźnik wiersza bieżącego (wiersz bieżący jest to wiersz, do którego

skierowane są kolejne operacje redagujące). Komendy modyfikujące położenie wskaźnika wiersza nazwane są komendami lokalizującymi; są one wymienione w tablicy 3.

Komendy lokalizujące pozwalają na:

- ✱ Ustawienie wskaźnika wiersza na początku lub na końcu zbioru wejściowego lub bufora blokowego.
- ✱ Przesunięcie wskaźnika wiersza o określoną ilość wierszy od bieżącej pozycji.
- ✱ Przesunięcie wskaźnika wiersza do wiersza zawierającego podany tekst.

W trybie wydawania komend znak powrotu karetki i ESC działają przeciwstawnie. Pojedynczy znak <CR>przesuwa wskaźnik wiersza o jeden wiersz wstecz. W każdym wypadku wiersz bieżący jest wyprowadzany na terminal.

Jeżeli opcja VERIFY ON jest aktualnie ustawiona, po wydaniu komend BOTTOM, END, PAGE FIND, PAGE LOCATE lub SEARCH & CHANGE wiersz zlokalizowany będzie wydrukowany.

Tablica 3.

Komendy lokalizujące edytora wierszowego.

Komenda	Format	Opis
BEGIN lub TOP	Begin lub Top	Ustawiają wskaźnik wiersza na wiersz poprzedzający wiersz początkowy w zbiorze (tryb wierszowy) lub buforze blokowym (tryb blokowy). W trybie wierszowym obie komendy tworzą kopie zbioru za każdym razem, gdy są wywołane. Komendy te są równoważne.

BOTTOM lub END	Bottom lub End	Ustawiają wskaźnik wiersza na ostatnim wierszu w zbiorze lub buforze. Komendy te są równoważne.
<ESCApe>	Klawisz ESC	Drukuje wiersz poprzedni i sprawia, że zostaje on nowym wierszem bieżącym (tylko w trybie blokowym).
FIND	[n] Find [tekst]	Wyszukuje w bloku bieżącym lub zbiorze wejściowym n-tego wystąpienia ciągu znaków "tekst" na początku wiersza, znajdującego się w zbiorze po wierszu bieżącym. Tekst musi mieć początek w kolumnie 1-ej. Komenda ustawia wskaźnik wiersza na zlokalizowanym wierszu. Lokalizuje n-te wystąpienie ciągu znaków "tekst". W trybie blokowym przeszukiwanie kończy się wraz z końcem bloku.
LOCATE	[n]Locate tekst	

NEXT	Next [n] lub Next [-n]	Ustawia nowy wiersz bieżący w odległości n wierszy od aktualnego wiersza bieżącego.
NEXT & PRINT	NP[n] lub NP[-n]	Ustawia i drukuje nowy wiersz bieżący.
PAGE	PAGe n	Wprowadza tryb blokowy. Wczytuje n-tą stronę do bufora blokowego. Jeżeli n jest mniejsze od numeru strony bieżącej, realizuje najpierw komendę TOF. Strony są ograniczone znakami wysuwu strony <FF>. Komenda jest dozwolona tylko w trybie blokowym.
PAGE FIND	[n] PFind tekst	Wyszukuje w kolejnych blokach n-tego wystąpienia ciągu znaków tekst. Tekst musi rozpoczynać się w pierwszej kolumnie. Komenda jest dozwolona tylko w trybie blokowym.
PAGE LOCATE	[n] PLocate tekst	Wyszukuje w kolejnych blokach n-tego pojawienia się tekst-u. Tekst może wystąpić w którymkolwiek miejscu wewnątrz wiersza. Komenda jest dozwolona tylko w trybie blokowym.
<RETURN>	Klawisz <CR>	Drukuję wiersz następny, który zostaje wierszem bieżącym;

SEARCH &
CHANGE

SC /TEKST1
/TEKST2[/]

powoduje wyjście z trybu wprowadzania. Lokalizuje ciąg znaków tekst1 i zamienia go na tekst2.

3. Komendy modyfikacji i manipulacji tekstem

Komendy modyfikacji i manipulacji tekstem umożliwia zmianę zawartości redagowanego tekstu. Tablica 4 zawiera listę komend znajdujących się w tej grupie.

Tablica 4.

Komendy EDI modyfikacji i manipulacji tekstem

Komenda	Format	Opis
ADD	Add tekst	Dodaje ciąg znaków tekst na koniec wiersza bieżącego.
ADD & PRINT	AP tekst	Dodaje tekst do wiersza bieżącego i drukuje wiersz wynikowy.
CHANGE	[n]Change[/] /TEKST1/TEKST2	Zamienia ciąg znaków tekst1 na tekst2 n-razy w wierszu bieżącym.
DELETE	Delete [n] lub Delete [-n]	Usuwa wiersz bieżący i n-1 wierszy następujących, jeśli n jest znakiem (+); usuwa n wierszy poprzedzających wiersz bieżący, jeśli n jest ze znakiem (-). Wartość [-n] działa tylko w trybie blokowym.
DELETE & PRINT	DP[n] lub DP[-n]	Działa tak samo, jak komenda DELETE,

ERASE	ERASE [n]	oprócz tego, że drukuje nowy wiersz bieżący. W trybie wierszowym usuwa wiersz bieżący.
FORM FEED	FF	W trybie blokowym usuwa blok bieżący w buforze i (n-1) bloków następujących. Ustawia znak wysuwu strony <FF> do bufora blokowego (używany w celu ograniczenia zakresu strony).
INSERT	Insert tekst	Wprowadza ciąg znaków tekst za wierszem bieżącym lub ustanawia tryb wprowadzania, jeżeli tekst nie został wyspecyfikowany.
LINE CHANGE	[n]LC/TEKST1 /TEKST2[/]	Zmienia wszystkie ciągi znaków tekst1, które wystąpiły w wierszu bieżącym (i n-1 wierszach następujących) na tekst2.
OVERLAY	Overlay[n]	Usuwa n wierszy, ustawia tryb wprowadzania i w miejsce wiersza lub wierszy pierwotnych wstawia nowy wiersz lub wiersze wprowadzane z terminala.

PASTE	Paste/TEKST1 /TEKST2[/]	Przeszukuje wszystkie, pozostałe jeszcze w zbiorze lub buforze blokowym wiersze, w celu znalezienia ciągu znaków tekst1 i zamiany na tekst2.
RETYPE	Retype tekst	Zamienia wiersz bieżący na podany tekst lub usuwa ten wiersz, jeżeli tekst nie został wymieniony.
TOP OF FILE	TOF	Powoduje powrót wskaźnika wiersza bieżącego na początek zbioru wejściowego oraz zachowuje wszystkie wcześniej redagowane strony.
UNSAVE	UNSave [specyfikacja- zbioru]	Wstawia wszystkie wiersze ze specyfikowanego zbioru w miejsce następujące po wierszu bieżącym. Jeżeli specyfikacja zbioru nie jest podana, jako jej domniemanie wykorzystana jest nazwa SAVE.TMP.

4. Makrokomendy

Komendy te umożliwiają zdefiniowanie, zapamiętanie, odwołanie się i wykorzystanie makrorozkazów. Makrorozkaz jest to seria komend edytora, które raz zdefiniowane mogą być wykonywane wielokrotnie przez podanie ich identyfikatora,

a więc wywoływane w o wiele bardziej skróconej postaci. Tablica 5 zawiera wszystkie komendy EDI dla obsługi makrorozkazów.

Tablica 5.

Makrorozkazy edytora wierszowego.

Komenda	Format	Opis
MACRO	MACRO x definicja	Definiuje makrorozkaz o numerze x. Maksymalną wartością dla x jest 3, czyli maksymalnie trzy makrodefinicje.
MACRO CALL	MCall	Pobiera makrodefinicję pamiętaną w zbiorze MCALL.;n.
MACRO EXECUTE	[n]Mx [a]	Powoduje wykonanie makrorozkazu o numerze x n-razy, podstawiając wartość argumentu liczbowego [a]. Wartość x może być równa 1, 2 lub 3.
MACRO IMMEDIATE	[a] <definicja>	Definiuje makrorozkaz i natychmiast ją n-razy wykonuje. Makrodefinicja zapamiętana jest z numerem 1.

5. Komendy wprowadzania do lub ze zbioru

Komendy wprowadzania/wyprowadzania sterują przesyłaniem tekstu do i ze zbioru oraz przechowują zbiory. Tablica 6 zawiera wszystkie tego typu komendy.

Tablica 6.

Komendy wejścia/wyjścia EDI.

Komenda	Format	Opis
FILE	FILE specyfikacja-zbioru	Przesyła wiersze ze zbioru wejściowego do dwu zbiorów: zbioru wyjściowego i zbioru specyfikowanego, aż do napotkania znaku wysuwu strony <FF> lub znacznika końca zbioru (tylko w trybie wierszowym).
READ	READ [n]	Wczytuje kolejnych n bloków tekstu ze zbioru wejściowego do bufora blokowego. Jeżeli bufor zawiera już jakiś tekst, nowy tekst jest do niego dodany.
RENEW	RENew [n]	Zapisuje blok bieżący do zbioru wyjściowego i wczytuje nowy blok ze zbioru wejściowego.
SAVE	SAVE [n][specyfikacja-zbioru]	Umieszcza wiersz bieżący i n-1 następujących wierszy w zbiorze wyspecyfikowanym. Jeżeli specyfikacja-zbioru nie została podana, wiersze umieszczone są w zbiorze SAVE.TMP. Ko-

menda SAVE umieszcza utworzony zbiór tymczasowy na tym samym urządzeniu i w tym samym UFD, w którym edytowany jest zbiór. Użytkownik może zmienić to domniemanie specyfikując określone urządzenie i określone UFD.

WRITE

Write

Zapisuje zawartość bufora blokowego do zbioru wyjściowego. Zawartość bufora jest usuwana.

6. Komendy wydruku

Komendy te kierują wydrukiem na terminalach lub na pseudo-urządzenie (CL:). Są one wymienione w tablicy 7.

Tablica 7

Komendy wydruku EDI.

Komenda	Format	Opis
LIST ON TERMINAL	List	Wprowadza na terminal wszystkie wiersze przechowywane aktualnie w buforze blokowym (tryb blokowy) lub zbiorze wejściowym (tryb wierszowy), począwszy od wiersza bieżącego.

LIST ON PSEUDO- DEVICE	LP	Działa tak samo jak komenda LI, poza tym, że wydruk jest wyprowadzany na pseudourządze - niu o mnemoniku CL:.
PRINT	Print [n]	Drukuje wiersz bieżący i n-1 wierszy następujących. Ostatni z wydrukowanych wierszy jest nowym wierszem bieżącym.
TYPE	Type [n]	Drukuje n kolejnych wierszy. W trybie wierszowym działa identycznie do komendy PRINT. W trybie blokowym wskaźnik wiersza pozostaje przy wierszu bieżącym, chyba że zostanie napotkany znacznik końca bloku EOB.

7. Komendy zamykania zbiorów

Komendy CLOSE i EXIT kończą działanie edytora wierszowego i zapisują resztę zbioru wejściowego do zbioru wyjściowego. W tablicy 8 wymienione są te komendy.

Tablica 8.

Komendy zamykania zbiorów.

Komenda	Format	Opis
CLOSE	Close [specyfikacja-zbioru]	Przesyła, pozostałe w buforze i zbiorze wejściowym wiersze do zbioru wyjściowego i zamy-

		ka te zbiory. Jeżeli wykorzystana została specyfikacja zbioru, zbiór wyjściowy będzie miał zmienioną nazwę. Po zakończeniu operacji zamykania zbiorów zostaje wyprowadzone przynaglenie edytora wierszowego:EDI>. Zamyka pomocniczy zbiór wejściowy.
CLOSE SECONDARY	CLOSES	
CLOSE & DELETE	CD1 [specyfikacja-zbioru]	Działa tak samo, jak CLOSE, poza tym, że dodatkowo usuwa zbiór wejściowy. Wyprowadzone zostaje przynaglenie EDI>.
CTRL/Z	Znak ^Z	Zamyka zbiory i kończy sesję edycji.
EXIT	EXit [specyfikacja-zbioru]	Zamyka zbiory, opcjonalnie zmienia nazwę zbioru wyjściowego i kończy sesję edycji.
EXIT & DELETE	EDx [specyfikacja-zbioru]	Przesyła, pozostawia w buforze i zbiorze wejściowym wiersze do zbioru wyjściowego oraz zamyka zbiory. Zmienia nazwę zbioru wyjściowego, w wypadku gdy podana jest specyfikacja

KILL

KILL

zbioru. Usuwa
zbiór wejściowy i
kończy działanie
edytora EDI.
Zamyka zbiory wej-
ściowy i wyjściowy,
usuwa zbiór wyjś-
ciowy. Wydane jest
przynaglenie EDI).

8. Komunikaty informacyjne na poziomie komendy

Komunikaty z tej grupy dają informację, która jest określona jako pomocnicza lub może służyć do rozpoznania błędów, które wystąpiły w poprzedniej komendzie. Wszystkie komunikaty tej grupy ograniczone są nawiasami kwadratowymi i następuje po nich przynaglenie dla nowej komendy w postaci *. Na przykład: jeśli komenda DELETE napotyka w trybie blokowym znacznik końca bloku, pojawi się następujący komunikat:

[*EOB*]

*

Poniżej opisane są wszystkie komunikaty należące do danej grupy.

[BUFFER CAPACITY BY]

Przekroczenie pojemności bufora

(wiersz przekraczający pojemność bufora)

[LINE DELETED]

(Wiersz został usunięty)

Komenda READ, UNSAVE, INSERT lub OVERLAY wywołała przekroczenie pojemności bufora blokowego. Wiersz, który spowodował przepełnienie został wyprowadzony i usunięty.

Jeśli tworzony jest nowy zbiór, należy opróżnić bufor przy pomocy komendy WRITE i kontynuować sesję edycji.

Jeśli redagowany jest zbiór już istniejący, może być możliwe dalsze działanie poprzez wydanie

komenda RENEW lub WRITE. W przeciwnym wypadku można wydać komendę CLOSE w celu zamknięcia zbioru wyjściowego i zachowania wszystkich wyników operacji redagujących. Następnie należy otworzyć zbiór wyjściowy jako zbiór wejściowy i użyć komendę SIZE w celu zmniejszenia ilości wierszy wczytywanych za każdym razem do bufora. Po czym należy używając komendy PAGE LOCATE, znaleźć miejsce w zbiorze, od którego redakcja tego zbioru ma być kontynuowana.

[CREATING NEW FILE]

INPUT

(Tworzenie nowego zbioru)

(Wprowadzanie)

Specyfikowany w wierszu komendy zbiór wejściowy nie istnieje i EDI tworzy nowy zbiór. EDI automatycznie ustawia tryb wprowadzania i czeka na wprowadzenie wierszy tekstu. Jeżeli użytkownik zamierzał utworzyć nowy zbiór, może kontynuować działanie wprowadzając nowe wiersze. W przeciwnym wypadku należy ustawić tryb wydawania komend, wprowadzając znak <CR>. Następnie wydać komendę KILL w celu usunięcia zbędnego nowego zbioru. Gdy EDI wyda przynaglenie dla nowej specyfikacji zbioru, należy wprowadzić poprawną specyfikację zbioru.

[ILL CMD]

(Nieprawidłowa komenda)

EDI albo nie rozpoznaje komendy albo komenda nie jest zgodna z aktualnie działającym trybem (np. komenda READ w trybie wierszowym).

[ILL NUM]

(Nieprawidłowa wartość)

W wypadku gdy zadana była wartość liczbowa, dostarczono znak nienumeryczny lub podano wartość ujemną, w wypadku gdy wymagana jest wartość dodatnia.

[ILL STRING CONST]

(Błąd ograniczników tekstu)

Specyfikowany w komendzie CHANGE, LC, PASTE lub SC "tekst" przeznaczony do odszukania nie zawiera skojarzonych znaków ograniczających go (np. PASTE/ALPHABETA zamiast prawidłowej postaci PASTE/ALPHA/BETA).

[ILLEGAL IN BLOCK ON MODE]

(Komenda nielegalna w trybie blokowym)

Komunikat pojawia się w wypadku próby wykonania komendy nielegalnej w trybie blokowym, takiej jak na przykład FILE lub OUTPUT ON/OFF.

[ILLEGAL FOLE NAME GIVEN IN CLOSE OR EXIT]

(Podana błędna nazwa zbioru przy komendzie CLOSE lub EXIT)

lub

[FILE WAS NOT RENAMED]

(Nazwa zbioru nie została zmieniona)

Nieprawidłowa syntaktycznie specyfikacja zbioru, podana przy komendzie CLOSE lub EXIT, odrzucona próba zmiany nazwy zbioru wyjściowego lub nie wykonana próba operacji EXIT lub CLOSE dla zmiany nazwy zbioru z umieszczeniem go na innym urządzeniu.

Można zmienić nazwę zbioru, jeśli to konieczne, wykorzystując program manipulacji zbiorami PIP.

[MACRO NOT DEFINED]

(Makrorozkaz nie był zdefiniowany)

Komunikat pojawia się w wypadku próby wykonania makrorozkazu przy pomocy komendy M, gdy określona makrorozkaz nie był zdefiniowany.

Należy użyć komendy MACRO w celu zdefiniowania makrorozkazu i wówczas wykonać go, wydając komendę M.

[MACRO NUMERIC ARG UNDEFINED]

(Argument liczbowy makrorozkazu nie został określony)

Próba wywołania makrorozkazu bez dostarczenia argumentu liczbowego. Makrodefinicja zawierała

znak % i w tym wypadku wymagane jest podanie argumentu numerycznego przy komendzie MACRO EXECUTE.

Należy powtórzyć komendę, specyfikując odpowiedni argument liczbowy.

[MCALL FILE DOES NOT EXIST]

(Zbiór MCALL nie istnieje)

Wydano komendę MCALL w celu zdefiniowania makrorozkazu, ale nie został odnaleziony zbiór MCALL. Zadany zbiór makrorozkazu może znajdować się w innym katalogu zbiorów użytkownika. W takim wypadku wystarczy używając programu PIP, skopiować zbiór lub zmienić nazwę zbioru MCALL w bieżącym katalogu.

[NO INPUT FILE OPEN]

(Zbiór wejściowy nie jest otwarty)

Wydano komendę PAGE lub RENEW podczas tworzenia nowego zbioru (a więc w wypadku gdy nie ma zbioru wejściowego). Komendy te mogą być wykonane tylko wtedy, gdy redagowany jest zbiór istniejący.

[NO MATCH]

(Brak znaku)

W bieżącym wierszu nie ma specyfikowanego tekstu, przeznaczonego do modyfikacji przy pomocy komendy CHANGE.

[OVERLAYING PREVIOUSLY DEFINED MACRO]

(Nałożenie poprzednio zdefiniowanego makrorozkazu)

Makrokomenda wywołała redefinicję poprzednio zdefiniowanego makrorozkazu. Komunikat ten oznacza, że poprzednia definicja nie jest już dłużej aktualna.

[SAVE FILE DOES NOT EXIST]

(Zbiór przechowywany nie istnieje)

Zbiór specyfikowany w komendzie UNSAVE nie został odnaleziony. Należy upewnić się co do prawidłowości podanej specyfikacji zbioru. Jeśli

specyfikacja zbioru nie była podana, komunikat ten oznacza, że uprzednio nie była wydana komenda SAVE (bez podania specyfikacji zbioru),

[SECONDARY FILE ALREADY OPEN]

(Zbiór pomocniczy jest jeszcze otwarty)

Dokonano próby otwarcia pomocniczego zbioru wejściowego, podczas gdy wciąż otwarty jest jeszcze inny zbiór pomocniczy. Sytuacja taka jest także możliwa w wypadku, gdy wadano komendę CLOSE lub KILL lub gdy wystąpił błąd, przy którym nieodwołalne było restartowanie EDI. Pierwszy przypadek interpretowany jest jako błąd, w drugim natomiast komunikat informuje tylko, że wciąż otwarty jest zbiór pomocniczy. Należy zamknąć pomocniczy zbiór wejściowy, używając komendy CLOSES i wówczas dopiero otworzyć zbiór pomocniczy za pomocą komendy OPENS.

[SECONDARY FILE CURENTLY SELECTED FOR INPUT]

(Zbiór pomocniczy wybrany jako wejściowy)

Wydano komendę CLOSE lub KILL albo wystąpił błąd powodujący restart EDI, w wypadku gdy pomocniczy zbiór wejściowy jest otwarty i wybrany jako wejściowy.

Należy wydać komendę SP, a następnie komendę CLOSES.

[SYNTAX ERROR]

(Błąd składni)

Wprowadzona komenda posiada nieprawidłową strukturę.

[TOO MANY CHARS]

(Zbyt dużo znaków)

Komunikat może pojawić się w wyniku działania komend CHANGE, IC, PASTE, SC lub ADD, gdy ilość wstawianych znaków przekracza wielkość wiersza. EDI dopuszcza wiersze o długości do 132 znaków.

Aby zapewnić sobie prawidłową postać wiersza, należy ponownie go przepisać, względniąc ograniczenia związane z jego długością.

[«BOB»]

(Początek bufora)

Komunikat sygnalizuje, że został osiągnięty początek bufora blokowego. Wskaźnik wiersza bieżącego jest spozycjonowany bezpośrednio przed pierwszym wierszem w buforze. Dzięki temu nowe wiersze tekstu mogą być wprowadzone przed pierwszym wierszem bufora.

[«EOB»]

(Koniec bufora)

Komunikat sygnalizuje, że został osiągnięty koniec bufora blokowego. Wskaźnik wiersza bieżącego umieszczony jest teraz na końcu bufora. W tym wypadku, jeśli wstawiane będą nowe wiersze, wówczas będą one zmieszczane po ostatnim wierszu tekstu, znajdującego się w buforze.

[«EOF»]

(Koniec zbioru)

Komunikat sygnalizuje, że został osiągnięty znacznik końca zbioru wejściowego. Jeśli wszystkie operacje w danej sesji edycji są zakończone, wówczas należy użyć komendy CLOSE lub EXIT w celu zamknięcia zbioru wyjściowego. W innym wypadku należy wydać komendę TOF, aby wrócić do pierwszego bloku w zbiorze.

PROGRAM TESTOWANIA URZĄDZEŃ ZEWNĘTRZNYCH IOX

1. Wstęp

Program IOX służy do testowania urządzeń we/wy w systemie DOC PB. Testy IOX uruchamia się za pomocą języka komend IOX. W wyniku działania IOX uzyskuje się szczegółowy raport błędów oraz ogólne informacje na temat sprawności systemu. IOX testuje jednostki w systemie w taki sposób, że zapisuje do tomu znajdującego się w danej jednostce dane wzorcowe (wzorzec), po czym odczytuje to, co zapisał. Następnie sprawdza, czy zachodzi zgodność pomiędzy wartością odczytaną i zapisaną. Wystąpienie różnic oznacza błąd. IOX raportuje błędy w momencie ich wykrywania. Wyświetla również raporty aktywności systemu: na zlecenie, cykliczne (w zadanych przedziałach czasu) oraz po zakończeniu działania.

IOX posiada język komend używany do kierowania funkcjami testowania we/wy. Są następujące grupy komend:

1. dołączanie jednostek do konfiguracji testowej
2. wybieranie określonych jednostek do testowania
3. ustalanie warunków wykonania testów we/wy
4. sterowanie wykonaniem testów
5. komendy przekazywane przez IOX do MCR, realizowane w trakcie testowania
6. mierzenie sprawności systemu, z użyciem jawowej pętli IOX

IOX obsługuje następujące urządzenia:

<u>Urządzenie</u>	<u>Symbol</u>
SMRP4/SMRP5/SMRP6	DB:
SMRF11	DF:
SMRK5	DK:
SMRL1/SMRL2	DL:
SMRK6/SMRK7	DM:
SMRP2/SMPR2/SMPR3	DP:

SMRM2/SMRM3
SMRS3/SMRS4
SMRX1
SMRX2
SMTU60
SMTU16/SMTE16/SMTU45/SMTU77
SMTS4
SMTU10/SMTE10/SMTS03

DR:
DS:
DX:
DY:
CT:
MM:
MS:
MT:

2. Zestawienie komend IOX

<u>Komenda</u>	<u>Skrócony opis</u>
ABORT	Przerywa wykonanie testu we/wy przed normalnym zakończeniem
ABORT task	Przerywa wykonanie zadania (task) zapoczątkowanego przez komendę EXECUTE
ABORT /ALL	Przerywa wykonanie zadań zapoczątkowanych komendą EXECUTE
BADBLOCKS	Dołącza numery uszkodzonych bloków do listy uszkodzonych bloków; wyświetla tę listę
BUFFERSIZE	Ustala domniemaną wielkość bufora używanego przez IOX do zapisu i odczytu, w przypadku testowania jednostki z tomem nie mającym struktury zbiorów DOC PB
CONFIGURE	Dołącza jednostkę do konfiguracji testowej; wyświetla listę jednostek konfiguracji testowej
CONTROL_C	Zezwala lub nie zezwala na przerywanie testu IOX w celu dialogowego wprowadzenia komendy
DENSITY	Ustala gęstość zapisu przy testowaniu taśmy magnetycznej; wyświetla bieżącą lub domniemaną gęstość zapisu oraz słowo charakterystyki
DESELECT	Kończy (przerywa) testowanie jednostki

ERRORLIMIT	Ustala liczbę błędów tolerowanych przez IOX, zanim jednostka będzie usunięta z konfiguracji testowej
EXECUTE	Skierowuje wiersz komendy do systemu operacyjnego w celu wykonania komendy
EXIT	Powoduje zakończenie działania IOX i powrót do systemu operacyjnego (MCR)
FILES11	Wybiera jednostkę z zamontowanym tomem o strukturze zbiorów DOC PB; nakazuje programowi IOX ochronę struktury danych tomu podczas testowania
HELP	Wyświetla zestawienie komend IOX na terminalu
INTERLEAVE	Ustala współczynnik opuszczenia zapisywanych/czytanych bloków dyskowych przy testowaniu dysków
ITERATIONS	Ustala ilość przebiegów jałowej pętli IOX, powodującą jednokrotną inkrementację licznika powtórzeń tej pętli
LOGFILE	Skierowuje raporty sprawności i raporty błędów na terminal albo do zbioru rejestracyjnego IOX.LOG
PARAMETERLIST	Wyświetla bieżące wartości parametrów IOX
PATTERN	Ustala wzorzec (dane wzorcowe), który IOX zapisuje i odczytuje podczas testowania; wyświetla wszystkie wzorce na terminalu
PRINTSUMMARY	Wyświetla na terminalu raport z działania IOX
PROCEED	Ustawia wykonanie testu IOX po przerwaniu go przez <CTRL/C> w trybie dialogowym
RANDOM	Zezwala lub nie zezwala na losowe testowanie bloków zamiast sekwencyjnego

REPORTERRORS	Zezwala lub nie zezwala na wy- prowadzanie raportów błędów na terminal
RESTART	Restartuje IOX: wstrzymuje tes- towanie, usuwa z konfiguracji testowej wszystkie jednostki oprócz domniemanych, określa wszystkie jednostki jako nie wy- brane do testowania, ustawia wszystkie parametry na początko- we wartości domniemane
RETRIES	Zezwala lub nie zezwala na po- wtarzanie operacji we/wy przez drajwery urządzeń, gdy wykrywa- ją błędy we/wy
RUNTIME SELECT	Ustala czas trwania testowania Wybiera do testowania we/wy jed- nostkę z tomem nie mającym stru- ktury zbiorów
SPY	Wyświetla informacje w czasie testowania dla określonej lub dla wszystkich jednostek
START	Rozpoczyna testowanie operacji we/wy dla wybranych jednostek konfiguracji testowej
SUMMARYTIME	Ustala przedział czasu pomiędzy cyklicznymi raportami
TEMPORARYFILE	Ustala wielkość zbioru robocze- go używanego przez IOX przy te- stowaniu jednostek z zamontowa- nymi tomami o strukturze zbiorów DOC PB
VERIFY	Upewnia, że dane mogą być z jed- nostki odczytywane
VOLUMECHECK	Sprawdza, czy jednostka, która ma być testowana, ma zamontowa- ny tom o strukturze zbiorów DOC PB
WAIT	Określa wykonywanie algorytmu jałowej pętli IOX

WRITECHECK Zezwala lub nie zezwala na kontrolę zapisu na dyskach.

3. Uruchamianie IOX

Jeżeli IOX jest w systemie zainstalowany, to przywołuje się go używając komendy: >IOX. IOX odpowiada wypisaniem przynaglenia: IOX>. Jeżeli IOX nie jest zainstalowane, to przywołuje się go, używając komendy RUN:

```
>RUN *IOX/INC=nn
```

gdzie nn oznacza po 1.2 K słów na każde testowane urządzenie. W przypadku zastosowania komendy RUN, IOX nazywa zadanie używając numeru terminala i wypisuje przynaglenie, np. TT36>. Ustalanie konfiguracji testowej. Przez domniemanie IOX włącza do konfiguracji testowej wszystkie jednostki z numerami zerowymi. Inne jednostki, jeżeli mają być testowane, należy również włączyć do konfiguracji. Na przykład:

```
IOX>CONFIGURE DK1:
```

```
IOX>CONFIGURE DK2:
```

```
IOX>CONFIGURE MM5:
```

Ustalanie parametrów wykonania. Można ustalać warunki wpływające na wykonywanie testów we/wy. Na przykład, można określić: przedział czasu testowania, częstotliwość raportów sprawności, czy raporty sprawności mają być skierowane do zbioru rejestracji SYO:IOX.LOG. Można także zlecić programowi IOX, aby sprawdzał, czy testowane jednostki mają zamontowane tomy o strukturze zbiorów DOC PB. Przykładami komend ustalania parametrów czasu wykonywania są:

```
IOX>RUNTIME 10
```

```
IOX>SUMMARYTIME 4
```

```
IOX>LOGFILE YES
```

Powyższe komendy nakazują IOX-owi testowanie operacji we/wy w ciągu 10 minut, wyprowadzanie raportów co 4 minuty i umieszczenie wszystkich raportów w zbiorze rejestrowym. Wybieranie jednostek do testowania operacji we/wy. Jeżeli już konfiguracja testowa jest okre-

ślona, to można wybierać jednostki, które mają być testowane. Testowane są tylko jednostki jawnie wybrane. Na przykład:

IOX>FILES11 DK1:

Ta komenda wybiera do testowania jednostkę DK1. Do wybierania jednostek do testowania służy zarówno komenda FILES11, jak i komenda SELECT. Komenda FILES11 używa się wtedy, gdy podczas testowania ma być chroniona struktura zbiorów DOC PB.

Inicjowanie testowania. Testowanie operacji we/wy rozpoczyna się po ustaleniu konfiguracji testowej i wybraniu jednostek do testowania, poprzez wprowadzenie komendy START:

IOX>START

IOX wykonuje wówczas np. test wzorca na DK1:, w czasie określonym komendą RUNTIME np. 10 minut. Po zakończeniu testowania IOX wyprowadza zestawienie końcowe, a wcześniej - raporty okresowe, skierowując je do zbioru rejestrowego IOX.LOG. Dostęp do tego zbioru uzyskuje się po wyjściu z IOX poprzez wprowadzenie komendy:
>PIP TI:=IOX.LOG.

Zakończenie pracy IOX. Służy do tego komenda EXIT: IOX>EXIT

4. Tryby działania IOX

IOX działa w trzech trybach: tryb komend, tryb wykonywania, tryb dialogowy. Trybami steruje się z klawiatury terminala, używając znaków <CTRL/C> i <CR>.

Zwykle w systemie DOC PB użycie <CTRL/C> powoduje przerwanie zadania i oddanie sterowania do programu MCR. Otrzymuje się wówczas na terminalu przynaglenie MCR). Natomiast w przypadku IOX znak <CTRL/C> może być stosowany w innym celu, jeżeli uprzednio wprowadzi się komendę CONTROL_C. W takim przypadku użycie <CTRL/C> powoduje przerwanie testowania operacji we/wy i oddanie sterowania do IOX (a nie do MCR).

Tryb komend. Bezpośrednio po przywołaniu IOX działa w trybie komend, oczekując na wprowadzenie komendy. W trybie komend naciśnięcie <CR> powoduje, że IOX wyprowadza przynaglenie: IOX>. Użycie <CTRL/C> powoduje wyprowadzenie przez system przynaglenia MCR> i mają wówczas zastosowanie normalne reguły wprowadzania komend MCR. W trybie komend można użyć komendy CONTROL_C w celu zezwolenia lub nie zezwolenia na przerwanie testowania operacji we/wy. CONTROL_C YES zezwala na przerwania w celu dialogowego wprowadzania komend; CONTROL_C NO zabrania przerywania testowania we/wy.

Tryb wykonania. Jeżeli testowanie operacji we/wy jest rozpoczęte (to znaczy była komenda START), to IOX wchodzi w tryb wykonania. Jeżeli obowiązuje zezwolenie na przerwania testowania (tzn. podano CONTROL_C YES), to IOX nie przyjmuje z terminala nic oprócz <CTRL/C>. Na próby wprowadzenia czegoś innego IOX odpowiada komunikatem:

IOX -- ONLY ^C ALLOWED

Natomiast, gdy przerwania są zabronione (przez komendę CONTROL_C NO), to użycie <CTRL/C> powoduje wyprowadzenie normalnego przynaglenia MCR>, zaś wykonanie IOX jest kontynuowane. Tak więc w tym przypadku testowanie nie może być przerwane. Jedynym sposobem przerwania testowania jest tu użycie komendy ABORT: >ABO IOX

Tryb dialogowy. Kiedy jest ustawione CONTROL_C YES, to testowanie operacji we/wy może być przerywane w celu dialogowego wprowadzania komend. Wówczas użycie <CTRL/C> powoduje wyprowadzenie przynaglenia: IOX>. Po otrzymaniu tego przynaglenia można wprowadzić dowolną komendę IOX, z wyjątkiem następujących: BUFFERSIZE, PATTERN (ale PATTERN bez argumentów jest dozwolone), RUNTIME, RESTART, START. Te komendy mogą być wprowadzane tylko w trybie komend. Użycie <CTRL/C> ustanawia tryb dialogowy, aż do wystą-

pienia komendy PROCEED. W trybie dialogowym testowanie jest kontynuowane, ale raporty błędów i raporty sprawności skierowane na terminal (jeżeli podano wcześniej LOGFILE NO) są zawieszane do momentu wystąpienia komendy PROCEED. Natomiast, jeżeli podano wcześniej LOGFILE YES, to raporty są normalnie cały czas wyprowadzane (do zbioru rejestrowego).

<CTRL/C> zatrzymuje zegar IOX-a, zatem czas nie jest wówczas mierzony, pomimo, że testowanie jest kontynuowane. W konsekwencji testowanie może trwać dłużej niż to określono w komendzie RUNTIME.

5. Ustawianie parametrów czasu wykonania

Wiele z komend IOX służy do ustalania warunków w czasie wykonania, takich, jak czas testowania, czas pomiędzy raportami okresowymi itd. Komenda PARAMETERLIST wyświetla wszystkie parametry czasu wykonania i ich aktualne wartości.

Parametry czasu wykonania dzielą się na 5 kategorii:

- ogólne warunki czasu wykonania
- warunki dla jednostek z tomami o strukturze zbiorów DOC PB
- warunki dla jednostek z tomami nie mającymi struktury zbiorów
- warunki dla mierzenia sprawności systemu
- wykonywanie innych zadań (oprócz IOX)

Ustalanie ogólnych warunków czasu wykonania.

Ogólne komendy czasu wykonania wpływają na przebieg wszystkich testów operacji we/wy. Przypuszczalnie użytkownik będzie chciał podawać te komendy przy każdym uruchomieniu IOX; można zatem umieścić je w zbiorze komend. A oto wykaz tych komend: LOGFILE, RUNTIME, SIMMARYTIME, ERRORLIMIT, REPORTERRORS, CONTROL_C, PATTERN.

Ustalanie warunków dla jednostek z tomami DOC PB. Aby ustalić warunki wykonania dla jednostek

z tomami o strukturze zbiorów DOC PB, używa się komend FILES11 i TEMPORARYFILE. Komenda FILES11 wybiera jednostki do testowania, natomiast komenda TEMPORARYFILE określa domniemany rozmiar zbioru roboczego, którego IOX używa w czasie testowania jednostki. To domniemanie (dotyczące rozmiaru) można zmienić komendą FILES11.

Ustalanie warunków wykonania dla tomów nie mających struktury zbiorów. W celu przetestowania jednostki z tomem nie mającym struktury zbiorów używa się komendy SELECT. Komenda SELECT nie sprawdza, czy tom ma strukturę zbiorów DOC PB; testuje cały tom, niszcząc wszystkie dane wcześniej zapisane. Aby upewnić się, czy wybrana jednostka nie zawiera tomu o strukturze zbiorów DOC PB, należy użyć komendy VOLUMECHECK (tzn. podać wcześniej VOLUMECHECK YES). Komenda VOLUMECHECK nakazuje IOX-owi ostrzec użytkownika w przypadku, gdy znajdzie struktury danych DOC PB w testowanym tomie.

Podczas testowania jednostki z tomem nie mającym struktury zbiorów można podać cztery warunki wykonania:

- domniemany rozmiar bufora - komenda BUFFER-SIZE
- metoda sprawdzania - komenda RANDOM lub INTERLEAVE
- umiejscowienie uszkodzonych bloków - komenda BADBLOCKS
- sposób wykonania operacji we/wy w czasie testowania - komendy: RETRIES i WRITECHECK

Ustalanie warunków dla mierzenia sprawności systemu. IOX przewiduje dwie komendy do mierzenia sprawności systemu: WAIT i ITERATIONS.

Równoległe wykonywanie innych zadań. Aby spowodować przekazanie komend do systemu, używa się komendy EXECUTE. Zadania tak uruchomione wykonują się równoległe z testowaniem operacji we/wy przez IOX. Na przykład:

```
EXECUTE ...BAD DK1:/LI
```

(Sytem operacyjny ma wykonać program pomocniczy BAD.)

6. Wydruki programu IOX

IOX generuje dwa typy wydruków: raporty aktywności i raporty błędów. Raporty aktywności są to zestawienia informacji o działaniu IOX, jak np. czas przebiegu, czas do końca przebiegu. Raporty błędów zawierają takie informacje, jak np. numer bloku, w którym wystąpił błąd operacji we/wy.

6.1. Raporty aktywności

Są trzy typy raportów aktywności:

- zestawienie końcowe - raport aktywności dotyczy całego przebiegu IOX
- raport okresowy - jest to raport podsumowujący działanie IOX w ostatnim przedziale czasu określonym komendą SUMMARYTIME
- raport na zlecenie - jest to raport podsumowujący działanie IOX w przedziale czasu pomiędzy ostatnim zestawieniem końcowym a momentem wprowadzenia komendy PRINTSUMMARY.

Używając komendy LOGFILE, można nakazać IOX-owi wyświetlać raporty końcowe, raporty okresowe i raporty błędów na terminalu bądź kierować je do zbioru IOX.LOG. Poniższe wiersze ilustrują ogólny format raportów aktywności IOX.

```
18-SEP-78 15:44:42 ##### EXERCISER STARTED ###  
TIME -- RUN: 1 MIN. ELAPSED: 1 MIN. REMAINING:  
0 MIN.
```

```
DKO: -- 1,183. REQUESTS TOTALING 1,183. BLOCKS  
WITH NO ERRORS
```

```
IDLE-LOOP COUNT: 724. ITERATIONS
```

```
18-SEP-78 15:45:56 ### EXERCISER TERMINATED ###
```

Wydruk raportu aktywności składa się z wierszy dotyczących zdarzeń i z wierszy informujących o stanie IOX. Wiersze dotyczące zdarzeń informują o tym, kiedy testowanie zostało rozpoczęte i zakończone, a także o takich zdarzeniach, jak przerwanie testowania, otwarcie zbioru rejestracji, zakończenie zapisów w tym zbiorze. Przykładowy wiersz opisujący zdarzenia ma następujący format:

18-SEP-78 15:45:56 *** EXERCISER STARTED ***
Wiersze informacji o stanie IOX dzielą się na 5 typów:

1. Informacje ogólne: data i czas raportowania
2. Informacje w minutach, czas pozostały do testowania
3. Informacje o jednostce: ilość zleceń operacji we/wy dla jednostki, ilość przetestowanych bloków, ilość wykrytych błędów
4. Informacje o jałowej pętli IOX, ilość iteracji tej pętli
5. Raporty błędów

6.2. Raporty błędów

Format raportu błędów wykrytych przez IOX:

DK1: -- UNRECOVERABLE ERROR AT BLOCK 358
(DECIMAL) 546 (OCTAL)
DK1: -- DATA ERROR AT BLOCK 358 (DECIMAL) 546
(OCTAL)
GOOD DATA: 165555 BAD DATA: 167155
(OCTAL)
WORD POSITION: 308 (DECIMAL) 464
(OCTAL)

Nie licząc pierwszego wiersza tego przykładu, raport zawiera trzy wiersze informacyjne. Pierwszy wiersz podaje numer bloku (dziesiętnie i oktalnie), w którym wystąpił błąd. W drugim wierszu znajduje się prawidłowa i błędna wartość słowa, w którym wystąpił błąd. Pozycja tego słowa w tym bloku jest podana w trzecim wierszu, dziesiętnie i oktalnie. Tylko pierwszy błąd wykryty w bloku jest raportowany.

7. Wyświetlanie informacji o stanie IOX

Są dostępne w tym celu cztery komendy: PARAMETERLIST, SPY, CONFIGURE, PATTERN.

Za pomocą komendy PARAMETERLIST uzyskuje się wyświetlenie wartości parametrów IOX w danym momencie. Poniżej podano format wyświetlania tych informacji.

<u>TASK</u>	<u>STATUS</u>	<u>COMMAND-LINE</u>		
...BAD	QUEUED	DKO:/LI		
<u>DEVICE</u>	<u>LUN</u>	<u>BUFSIZE</u>	<u>TYPE</u>	<u>FILESIZE</u>
DB1:	5.	512.	F11	500.
DB2:	6.	512.	F11	500.
MM1:	7.	1024	NFS	

BUFFERSIZE= 1024.	CONTROL_C= YES
ERRORLIMIT= 10.	LOGFILE= NO
INTERLEAVE= 4.	RANDOM= YES
ITERATIONS= 10.	REPORTERRORS= YES
PATTERN= 4.	RETRIES= NO
RUNTIME= 5.	VOLUMECHECK= YES
SUMMARYTIME= 1.	WAIT= YES
TEMPORARYFILE= 500.	WRITECHECK= NO

DYNAMIC MEMORY USAGE= 3278.:2248.:2248.:1.

Dla zadań zainicjowanych komendą EXECUTE komenda PARAMETERLIST wyświetla:

- nazwę zadania zainstalowanego (...BAD)
- stan zadania (QUEUED, ACTIVE, SUCCESS, WARNING, ERROR, SV ERROR)
- wiersz wprowadzonej komendy (DKO:/LI)

ACTIVE oznacza, że IOX jest w trakcie testowania jednostki. Pozostałe oznaczenia stanu odnoszą się do dyrektywy egzekutora EXSTA (EXIT WITH STATUS). Są one opisane w podręczniku "Egzekutor".

Dla jednostek wybranych do testowania komenda PARAMETERLIST powoduje wyświetlenie następujących informacji:

- specyfikacja urządzenia (DB1:, DB2: i MM1:)
- numer logiczny jednostki (LUN) (5, 6 i 7)
- rozmiar bufora używanego przez IOX do odczytu i zapisu (512(10) dla jednostek z tomami o strukturze zbiorów DOC PB; 1024(10) dla jednostek z innymi tomami)
- typ tomu zamontowanego w danej jednostce (F11 dla tomów DOC PB; NFS dla tomów o innych strukturach; VFY dla tomów w trakcie weryfikacji)

- rozmiar zbioru roboczego używanego przez IOX do przetwarzania, jeżeli w jednostce zamontowany jest tom o strukturze zbiorów DOC PB
Wiersz opisujący dynamiczne wykorzystanie pamięci ma postać:

pam-całkowita:wolny-blok:wolne-bajty:fragmenty
pam-całkowita - rozmiar dynamicznej pamięci IOX, w bajtach.

wolny-blok - rozmiar największego wolnego bloku, w bajtach.

wolne-bajty - całkowita ilość wolnych bajtów dynamicznej pamięci IOX.

fragmenty - ilość fragmentów w dynamicznej pamięci IOX.

Komendy: SPY, CONFIGURE, PATTERN są opisane w p. "Opisy komend IOX".

8. Sposób użycia IOX ze zbiorami komend

W trybie komend można dla IOX stosować zbiory komend. IOX dopuszcza tylko jeden poziom zbiorów komend. Zbiory komend mogą być wykorzystane do ustawienia wartości parametrów. W wierszach poszczególnych komend można umieszczać komentarze. Załączmy, że zbiór RUN10.IOX zawiera następujące komendy ustawiania parametrów:

```
LOGFILE YES  
RUNTIME 10  
SUMMARYTIME 1  
CONTROL_C NO
```

Aby wykonać ten zbiór komend, należy napisać:
@RUN10.IOX

9. Opisy komend IOX

Konwencje notacji

Elementy wiersza komendy

Komendy IOX składają się z wierszy komendy i kluczy. Są dwa typy kluczy: typ numeryczny i typ logiczny. Dla typu numerycznego podaje się wartości liczbowe. Klucze typu logicznego wy-

magają jednej z wartości YES/NO. Wszystkie wartości liczbowe są wprowadzane jako dziesiętne, z wyjątkiem numerów jednostek urządzeń i numerów wersji zbioru.

Skracanie zapisu komend

Jeżeli jest dopuszczalna skrócona forma zapisu komendy lub klucza, wówczas skrót jest zapisywany (w opisie komendy) dużymi literami, a opcjonalne rozszerzenie skrótu - małymi. Poniższy przykład ilustruje, jak zapisano możliwość skrócenia komendy LOGFILE YES:

LOGfile Yes

Ten zapis oznacza, że poprawnymi specyfikacjami tej komendy są: LO, LOG, LOGF, LOGFI, LOGFIL, LOGFILE oraz Y, YE, YES.

Opisy komend w porządku alfabetycznym

Abort [taskname]

[/ALL]

Komenda ABORT powoduje zakończenie testowania operacji we/wy i zakończenie zadań zainicjowanych komendą EXECUTE.

BADblocks ddnn: [numer-bloku[:liczba]],...

Komenda BADBLOCKS pozwala na wprowadzenie listy uszkodzonych bloków w jednostce, która ma być testowana przez IOX. Pozycja "liczba" oznacza ilość kolejnych uszkodzonych bloków, poczynając od bloku o numerze określonym przez numer-bloku. Jak wynika z formatu, można umieszczać na liście także pojedyncze uszkodzone bloki. A oto przykład:

BADBLOCKS DK2: 100:2,3,300:100

Ta komenda informuje IOX o następujących uszkodzonych blokach: 100, 101, 3, 300, ..., 399.

Jeżeli zastosujemy komendę BADBLOCKS nie podając listy bloków, to IOX wyświetli numery bloków uszkodzonych, które ma na liście. IOX wykorzystuje tylko listę dostarczoną komendą BADBLOCKS.

Buffersize bajty

Komenda BUFFERSIZE zmienia rozmiar bufora wykorzystywanego przez IOX podczas testowania tomów nie mających struktury zbiorów DOC PB. IOX zapamiętuje podaną wielkość bufora w bajtach, zaokrąglając w górę do najbliższej pełnej czwórki. W czasie testowania IOX pobiera pamięć na bufory o zadanym rozmiarze dynamicznie tj. w miarę potrzeb. Komenda ta jest niedozwolona dla jednostek będących w trakcie testowania.

CONFigure [ddnn:]

Komenda CONFIGURE wprowadza jednostkę do konfiguracji testowej. Jednostki muszą być wprowadzone do konfiguracji, zanim mogą być wybierane do testowania komendami SELECT lub FILES11. Komenda umożliwia również wyświetlenie listy jednostek konfiguracji testowej.

CONFIGURE z kluczem ddnn: wprowadza daną jednostkę do konfiguracji testowej. Na przykład:

CONFIGURE DK2:

CONFIGURE bez klucza listuje wszystkie urządzenia konfiguracji testowej.

IOX)CONFIGURE

DBO: DB1: DB2: DFO: DKO: DK2: DLO: DMO:
DPO: DRO: DSO: DXO: DYO: CTO: MMO: MSO:
MTO:

Wszystkie jednostki z numerem zero są zaliczane do konfiguracji testowej przez domniemanie.

CONTRol_C [Yes]

[No]

Komenda CONTROL_C zezwala lub nie zezwala na używanie <CTRL/C> w celu przerwania testowania operacji we/wy. CONTROL_C YES zezwala na przerwania, natomiast CONTROL_C NO nie zezwala na przerwania (zabrania przerwania).

Szczegóły na temat tej komendy podano w p. "Tryby działania IOX".

DENSity dduu: [nnnn]

Komenda DENSITY ustanawia gęstość zapisu na taśmie magnetycznej w czasie testowania przez IOX. IOX zapamiętuje tę wielkość w słowie charakterystyki urządzenia.

Zestaw dopuszczalnych wartości nnnn zależy od typu taśmy. Dla taśmy MT: są dopuszczalne gęstości: 200, 556, 800 BPI. Dla taśmy MM: można wybrać gęstość 88 albo 1600 BPI, natomiast dla taśmy MS: jedyną dopuszczalną wartością jest 1600 BPI. Stąd wynika konieczność umieszczenia w wierszu komendy pozycji dduu: . Na przykład:

IOX)DENS MT: 556

Użycie komendy DENSITY bez pozycji nnnn powoduje, że IOX wyświetla bieżącą wartość zapisu dla podanego typu taśmy.

DESelect dduu:

Komenda DESELECT kończy testowanie podanego urządzenia. Ponadto usuwa listę uszkodzonych bloków związaną z tym urządzeniem, unieważnia przypisanie LUN dla danej jednostki. Jednostka pozostaje w konfiguracji testowej, ale staje się dostępna dla systemu operacyjnego.

ERRorlimit liczba

Komenda ERRORLIMIT ustala ilość błędów pomiędzy raportami okresowymi, jaką toleruje IOX, zanim przerwie testowanie jednostki z powodu zbyt wielu błędów.

EXecute nazwa-zadania wiersz-komendy

Komenda EXECUTE przekazuje wiersz komendy do wykonania przez system operacyjny. Komendę EXECUTE można stosować w trybie komend lub w trybie dialogowym. W obydwu tych przypadkach wyniki kierowane przez uruchomione zadanie nazwa-zadania na terminal są normalnie wyświetlane, niezależnie od tego, czy testowanie odbywa się czy nie.

Parametry wiersza komendy EXECUTE podlegają regułom wprowadzania komend w systemie DOC PB i regułom stosowania dyrektywy SPWNk.

EXIT

Komenda EXIT powoduje zakończenie działania IOX i przekazanie sterowania do systemu operacyjnego.

FILES11 dduu: rozmiar-zbioru

Komenda FILES11 wybiera do testowania operacji we/wy jednostkę w tomie o strukturze zbiorów DOC PB. Jednostka ta musi być wcześniej włączona do konfiguracji za pomocą komendy CONFIGURE. Komenda FILES11 spełnia te same funkcje, co komenda SELECT, z tym, że FILES11 stosuje się do jednostek z tomami o strukturze zbiorów DOC PB, zaś SELECT - do jednostek z tomami nie mającymi tej struktury.

Przykład użycia komendy FILES11:

```
FILES11 DK1: 1000
```

Ta komenda powoduje wybranie jednostki DK1: do testowania operacji we/wy. Liczba 1000 zastępuje wcześniejszą wartość rozmiaru zbioru roboczego stosowanego przez IOX przy testowaniu jednostek z zamontowanymi tomami DOC PB. Rozmiar tego zbioru ustala się komendą TEMPORARYFILE.

Help

Komenda HELP wyświetla na terminalu skrócony opis komend IOX.

INTERleave liczba

Komenda INTERLEAVE określa współczynnik opuszczania bloków dyskowych na testowanych jednostkach (wyrażony liczbą bloków).

Zapisując wzorce testowe na dysku, IOX może zapisywać je zarówno do bloków wybieranych losowo, jak i do określonej sekwencji bloków. Aby nakazać IOX-owi testowanie losowe, dajemy komendę:

```
RANDOM YES
```

Aby nakazać IOX-owi testowanie określonej sekwencji bloków, należy podać RANDOM NO oraz określić liczbę bloków, które mają być opuszczone przed wykonaniem kolejnego zapisu wzorca.

Na przykład:

RANDOM NO

INTERLEAVE 2

W tym przykładzie współczynnik opuszczania jest ustawiony na 2, zatem IOX będzie zapisywał i odczytywał co drugi blok na dysku, poczynając od bloku 0.

Iterations liczba

Komenda ITERATIONS określa czas trwania jednej iteracji, liczony w powtórzeniach jałowej pętli IOX (liczba). Na przykład:

ITERATIONS 10

IOX zalicza jedną iterację na każde 10 powtórzeń jałowej pętli.

Komenda ITERATIONS jest stosowana wraz z komendą WAIT do mierzenia sprawności systemu.

Logfile [Yes]

[No]

Dzięki komendzie LOGFILE jest możliwe kierowanie raportów aktywności i raportów błędów zarówno na terminal, jak i do zbioru rejestracji SYO:IOX.LOG. LOGFILE YES powoduje skierowanie raportów do zbioru rejestracji, natomiast LOGFILE NO - na ekran terminala. Wartością domniemaną komendy jest LOGFILE NO.

Inne rodzaje wydruków (np. otrzymywane po komendach PRINTSUMMARY i SPY są zawsze kierowane na terminal.

Parameterlist

Komenda PARAMETERLIST wyświetla wartość parametrów wykonania IOX w momencie wydania komendy. Jest również wyświetlana informacja o aktualnie wybranych urządzeniach, aktywnych zadaniach oraz o ilości użytej pamięci dynamicznej. Przykład działania tej komendy podano w p. "Wyświetlanie informacji o stanie IOX".

PATtern [numer-wzorca]

Komenda PATTERN określa numer wzorca, który IOX zapisuje i odczytuje podczas testowania. Jest

11 wzorców, ponumerowanych w sposób następujący:

<u>Numer</u>	<u>Wzorzec</u>
1	-- 000000,000000
2	-- 177777,177777
3	-- 125252,052525
4	-- 165555,133333*
5	-- 163126,163126
6	-- 055555,155555
7	-- 022222,122222
10	-- 111111,111111
11	-- 007417,007417
12	-- 021042,021042
13	-- 104210,104210

Na przykład:

PATTERN 3

Powyższa komenda nakazuje użyć do testowania wzorzec 125252,052525. IOX oznacza gwiazdką wybrany wzorzec, jeżeli listuje wszystkie wzorce. Jeżeli brak komendy PATTERN, to domniemanym numerem wzorca jest 4.

PRIntsummary

Komenda PRINTSUMMARY drukuje informacje o testowaniu za okres od ostatniego raportu okresowego do momentu wprowadzenia komendy. Są one wyświetlane w takiej postaci, jak zestawienie końcowe. W p. "Wydruki programu IOX" pokazano format tego raportu.

PROceed

Komenda PROCEED służy do wznowienia wyświetlania raportów aktywności i raportów błędów na terminalu oraz do wznowienia działania zegara IOX po tym, jak funkcje te zostały zawieszane przez <CTRL/C>.

<CTRL/C> zawieszają raporty aktywności i raporty błędów oraz zatrzymują zegar IOX. PROCEED wznowia raportowanie na terminal i wznowia działanie zegara.

Random [YES]
[No]

Komenda RANDOM nakazuje IOX-owi użycie generatora liczb losowych do wyboru bloku do testowania. RANDOM YES powoduje użycie generatora liczb losowych; RANDOM NO powoduje, że IOX wykorzystuje współczynnik opuszczania, określony komendą INTERLEAVE.

Wartością domniemaną jest RANDOM YES.

REPorterrors [Yes]
[No]

Komenda REPORTERRORS zezwala lub nie zezwala na wyprowadzenie raportów błędów. REPORTERRORS YES nakazuje wyprowadzając raporty błędów; REPORTERRORS NO zabrania ich wyprowadzania. Wartością domniemaną jest REPORTERRORS YES. Raporty błędów mogą być wyświetlane na terminalu bądź umieszczane w zbiorze IOX.LOG, zależnie od ustawienia komendy LOGFILE.

REStart

Komenda RESTART nakazuje unieważnić wybór do testowania i przywiązanie (do zadania IOX) wszystkich jednostek, usunąć z konfiguracji testowej jednostki włączone przez użytkownika, ustawić wszystkie parametry na wartości domniemane, przywrócić zadaniu IOX pierwotny przydział pamięci.

RETriEs [Yes]
[No]

Komenda RETRIES nakazuje lub zakazuje drajwerowi we/wy powtarzać operacje we/wy w przypadku błędnego pierwszego wykonania operacji. RETRIES używa się przy testowaniu jednostek z tomami nie mającymi struktury zbiorów DOC PB.

RUNtime minuty

Komenda RUNTIME określa w minutach czas testowania jednostki lub jednostek. Jeżeli określi się RUNTIME 0, to testowanie będzie odbywać

się ciągle. Jeżeli brak komendy RUNTIME, to domniemany czas testowania wynosi 5 minut.

Select dduu: [rozmiar-bufora]

Komenda SELECT służy do wybierania do testowania operacji we/wy jednostek z tomami nie mającymi struktury zbiorów. Przykładem komendy SELECT jest:

```
SELECT DK1: 2000
```

Jednostka wybierana komendą SELECT musi być wcześniej umieszczona w konfiguracji przy pomocy komendy CONFIGURE.

Komenda SELECT zezwala IOX-owi na zniszczenie struktur danych zamontowanego tomu. Aby upewnić się, że nie jest to tom o strukturze zbiorów DOC PB, należy podać wcześniej komendę:

```
VOLUMECHECK YES
```

Jeżeli parametr rozmiar-bufora komendy SELECT zostanie pominięty, to jest przyjmowany rozmiar bufora z ostatniej komendy BUFFERSIZE.

Domniemana wielkość bufora wynosi 1024 bajty.

SPY [dduu:]

Komenda SPY wyświetla informację opisującą bieżący stan jednostki lub jednostek. Jeżeli nie określi się parametru dduu:, to jest wyświetlana informacja o wszystkich jednostkach.

SPY wyświetla dane w następującym formacie:

```
DBO: -- TYPE=F11 VBN=305. I/O AST QUEUED
```

```
FUNCTION=WRITE
```

Wiersz ten zawiera następujące informacje:

- nazwa testowanej jednostki (DBO:)
- bieżący testowany blok wirtualny (305(10))
- typ tomu zamontowanego w tej jednostce (F11)
- stan zleceń operacji we/wy: AST QUEUED, IN PROGRESS albo IDLE
- typ funkcji we/wy będącej w trakcie wykonywania (WRITE)

Start

Komenda START powoduje rozpoczęcie testowania operacji we/wy, w jednostkach wybranych komendami FILES11, SELECT.

SUmmarytime [minuty]

Komenda SUMMARYTIME określa przedział czasu pomiędzy raportami okresowymi. Wartość minuty = 0 oznacza, że raporty okresowe nie mają być wydawane. Wartością domniemaną tego parametru jest 1 minuta.

TEmporaryfile [rozmiar-zbioru]

Komenda TEMPORARYFILE jest stosowana wtedy, gdy chcemy testować jednostkę z zamontowanym tomem o strukturze zbiorów DOC PB. Określa rozmiar (w blokach) zbioru roboczego używanego przez IOX w czasie testowania takiego urządzenia. Zbiór roboczy jest oznaczony jako przeznaczony do usunięcia. Dzięki temu struktura zbiorów testowanego tomu nie jest zmieniona po testowaniu, nawet wtedy, gdy IOX kończy działanie przedwcześnie.

VERify ddu: [liczba-bloków]

Komenda VERIFY jest stosowana dla jednostek z zamontowanymi tomami nie mającymi struktury zbiorów. VERIFY upewnia, czy urządzenie prawidłowo wykonuje operacje we/wy, poprzez odczytywanie pewnej liczby bloków (liczbę tę wyznacza parametr liczba-bloków). Wykonując komendę VERIFY, IOX nie dokonuje testu wzorca. Nie jest to więc testowanie operacji we/wy.

VOLUMEcheck [Yes] [No]

Komenda VOLUMECHECK nakazuje (VOLUMECHECK YES) lub zabrania (VOLUMECHECK NO) sprawdzać, czy testowana jednostka zawiera tom o strukturze zbiorów DOC PB, tj. czy tom zawiera blok główny. W przypadku, gdy sprawdzenie jest nakazane (odbywa się to później, po komendzie START) i IOX znajdzie blok główny, wówczas wydaje komunikat ostrzegający, że jednostka zawiera tom o strukturze zbiorów DOC PB, a zatem mogą być zniszczone jakieś użyteczne informacje. Wartością domniemaną jest VOLUMECHECK YES.

WAIT [Yes]
[No]

Komenda WAIT jest używana wraz z komendą ITERATIONS do mierzenia sprawności systemu.

Podczas oczekiwania, aż jednostki zakończą operacje we/wy, IOX wykonuje jałową pętlę. Odbywa się to na jeden z dwóch sposobów:

1. Algorytm oczekiwania na semafor
2. instrukcja BICB - (SP), (SP)+

Instrukcja BICB pochłania ilość czasu CPU i może być wykorzystywana do mierzenia czasu całej operacji we/wy.

Algorytm oczekiwania na semafor jest wybierany przez domniemanie, ale można zabronić jego wybrania i wykorzystywać instrukcje BICB przez wprowadzenie komendy:

WAIT NO

Można dokładnie określić szybkość wykonania instrukcji BICB w danym procesorze, gdyż jest ona podana w odpowiednim podręczniku procesora. Mając te informacje oraz biorąc pod uwagę ilość wykonań tej instrukcji przypadającą na pojedynczą inkrementację licznika powtórzeń jałowej pętli IOX (co ustala komenda ITERATIONS), można oszacować czas trwania operacji we/wy w danym systemie.

WRITECHECK [Yes]
[No]

Komenda WRITECHECK zezwala lub nie zezwala drajwerom we/wy na kontrolę zapisu w czasie testowania operacji we/wy.

Komendy używa się do testowania jednostek z to-
mami nie mającymi struktury zbiorów. WRITECHECK
YES zezwala drajwerowi na kontrolę zapisu, zaś
WRITECHECK NO nie zezwala na kontrolę zapisu.

W czasie wykonywania operacji we/wy drajwer wy-
konuje 3 rozkazy: WRITE, WRITE CHECK i READ.
Poprzez wyeliminowanie WRITE CHECK uzyskuje się
zwiększenie szybkości testowania.

Wartością domniemaną jest WRITECHECK NO.

PROGRAM KONWERSJI ZBIORÓW FLX

Program konwersji zbiorów FLX jest pomocniczym programem przenoszenia zbiorów z jednego tomu drugiego. Dodatkowo przy przenoszeniu zbiorów między tomami o różnych strukturach następuje konwersja formatu zbioru przesyłanego na format zgodny ze strukturą tomu, na którym zbiór jest przesyłany.

FLX umożliwia:

1. Listowanie katalogów tomów AMKO albo tomów DOS-SM
2. Usuwanie zbiorów z tomów o strukturze DOS-SM i AMKO
3. Inicjowanie tomów AMKO i tomów DOS-SM

FLX dokonuje przenoszenia zbiorów i jeżeli to konieczne, zmianę formatu między:

tomami DOS-SM a tomami DOC-PB
tomami DOC-PB a tomami DOS-SM
tomami DOS-SM a tomami DOS-SM
tomami DOC-PB a tomami AMKO
tomami AMKO a tomami AMKO
tomami AMKO a tomami DOC-PB

Dopuszczalnymi urządzeniami DOS-SM są:

<u>Symbole urządzeń</u>	<u>Nazwa urządzenia</u>
CT	Kasetowa taśma magnetyczna SMTA
DK	Dysk SMRK
MM	Taśma magnetyczna SMTJU
MT	Taśma magnetyczna SMTM
PP	Perforator taśmy papierowej SMPC
PR	Czytnik taśmy papierowej SMPC lub SMPR

Dopuszczalnymi urządzeniami dla AMKO są:

<u>Symbole urządzenia</u>	<u>Nazwa urządzenia</u>
DK	Dysk SMRK
DL	Dysk SMRL1
DM	Dysk SMRK6 lub SMRK7
DX	Dysk elastyczny SMRX
DY	Dysk elastyczny SMRX2

FLX umożliwia korzystanie ze wszystkich urządzeń o strukturze DOC-PB, włącznie z taśmą kasetową. Tomy o strukturze DOC-PB są tomami domniemanymi. Tomy te inicjuje komenda monitora INITVOL. Tomy o strukturze AMKO i DOS-SM są inicjowane przez FLX. Obowiązują następujące klucze dla trzech formatów:

/RS dla DOC-PB
/DO dla DOS-SM i
/RT dla tomów AMKO.

1. Wiersz komendy FLX

Chociaż formaty poszczególnych funkcji FLX różnią się między sobą, ogólna forma zapisu komend FLX jest następująca:

devspec/sw=infilel/sw,...,infilen/sw
devspec Urządzenie określone dla FLX jako wyjściowe, które przyjmuje formę:
dev:[ufd]

Pole [ufd] jest nieobowiązkowe. Jeżeli nie jest określone, FLX używa bieżącego UIC. Jeżeli jawnie podano urządzenie wyjściowe, należy wprowadzić znak równości. FLX nie zezwala na specyfikacje zbioru wyjściowego. Zbiory wyjściowe przyjmują nazwy zbiorów wejściowych.

infilen Specyfikacje zbiorów wejściowych
/sw Jeden z trzech rodzajów kluczy FLX opisanych poniżej

FLX umożliwia stosowanie 9-znakowych nazw dla zbiorów DOS-SM na taśmie magnetycznej. Konwencja gwiazdki jest dozwolona tylko przy specyfikacji zbioru wejściowego. Numery wersji są dozwolone tylko dla zbiorów DOC-PB i nie mogą być określone przez konwencje gwiazdki.

2. Klucze FLX

FLX posiada trzy rodzaje kluczy dla operacji na zbiorach.

1. Klucze formatu tomu

2. Klucze formatu zbioru

3. Klucze sterujące

Klucze formatu tomu określają format tomów, na których zbiory są przechowywane tzn. DOC-PB, DOS-SM, AMKO.

Klucze formatu zbioru pozwalają na specyfikowanie formatu dla tomów innych niż DOC-PB. Zbiory mogą być formatowane znakowo, binarnie lub nieformatowane.

Klucze sterujące pozwalają na wybór rodzaju i parametrów operacji na zbiorach. Stosując klucze sterujące można przykładowo podać ilość bloków przypisanych do zbioru wyjściowego lub numer katalogu UFD dla zbioru wyjściowego.

2.1. Klucze formatu tomu

FLX posiada trzy rodzaje kluczy formatu tomu definiujące typ formatu tomu.

Klucz /DO - określa tom struktury DOS-SM

Klucz /RS - określa tom struktury DOC-PB

Klucz /RT - określa tom w formacie AMKO

Początkowo domniemywa się, że tomy wejściowe są w formacie DOS-SM a wyjściowe w formacie DOC-PB. FLX przyjmuje takie domniemania jeśli w wierszu komendy nie są podane określone klucze formatu tomu.

Przykłady:

FLX>DKO:=DXO:SYS1.MAC/RT - (Domniemane wyjście /RS)

FLX>DKO:/RT=DKO:SYS1.MAC - (Domniemane wejście /RS)

2.2. Klucze formatu zbioru

FLX ma trzy klucze formatu zbioru. Zbiory mogą być formatowane znakowo, binarnie albo nieformatowane. Konwersje formatu mogą być w dowolnym kierunku pomiędzy DOS-SM i DOC-PB albo pomiędzy AMKO i DOC-PB. Specyfikacja klucza formatu zbioru dotyczy formatu zbioru wyjściowego po konwersji.

Poniższa tabela opisuje klucze formatu zbioru.

KLUCZ

OPIS DZIAŁANIA

/FA:n

Format znakowy.

Zbiory wyjściowe DOC-PB, AMKO są formatowane znakowo. Zbiór znakowy jest zdefiniowany jako rekordy zawierające dane w kodzie ASCII zakończone znakami powrotu karetki oraz wysuwu papieru (SR-LF), lub znakiem przejścia do nowej strony (FF) lub znakiem pionowej tabulacji (VT). Podczas konwersji z DOS-SM albo AMKO do zbiorów DOC-PB pary CR-LF są usunięte z końca rekordów. W konwersji z DOC-PB do DOS-SM czy AMKO para CR-LF jest podana do końca każdego rekordu, który nie był zakończony LF lub FF. W obydwu kierunkach wszystkie znaki <O>, i <VT> są usuwane z rekordu wejściowego.

Jeśli klucz /FA:n jest specyfikowany dla zbiorów wyjściowych DOC-PB generowane są rekordy o stałej długości "n". Wyjściowe rekordy są uzupełnione zerami, jeśli to jest potrzebne. Jeśli nie określimy klucza /FA:n na wyjściu struktury DOC-PB, FLX tworzy rekordy o zmiennej długości. Rozmiar rekordu wyjściowego równy jest rozmiarami rekordu wejściowego. Dane znakowe są przesyłane jako 7-bitowe wartości. Bit ósmy każdego bajtu jest zerowany przed przesłaniem. CTRL/Z (w kodzie ASCII 32(8)) potraktowany jest jako logiczny koniec zbioru wejściowego dla przesłań w formacie znakowym z taśmy kasetowej albo taśmy perforowanej DOS-SM do DOC-PB.

/FB:n

Format binarny.

Zbiór wyjściowy w formacie DOS-SM lub AMKO będzie miał format binarny. W tym trybie nagłówki i sumy kontrolne są do-

dawane do rekordów wyjściowych w DOS-SM. albo AMKO i usuwane, podczas przekształcania tych zbiorów na strukturę zbiorów DOC-PB. Jeśli klucz /FB:n zostanie określony przy zbiorach wyjściowych DOC-PB wówczas rekordy wyjściowe są stałej długości (512(10) bajtów jest wielkością max.), określonej przez "n". FLX uzupełnia rekordy zerami w celu uzyskania określonej długości. Jeśli klucz /FB:n nie zostanie określona w strukturze zbiorów wyjściowych DOC-PB, FLX tworzy rekordy zmiennej długości. Wielkość rekordu wyjściowego równa jest długości rekordu wejściowego.

/IM:n Przesłanie będzie wykonywane bez formatowania.

Tryb obrazu zadania wymusza stałą długość rekordów. Można użyć wartości "n" w celu wskazania wcześniej ustalonej długości rekordu (w bajtach dziesięt - nie) dla zbiorów wyjściowych w strukturze DOC-PB (512 (10) bajtów - max. długość rekordu). Jeśli nie określono "n", to FLX przyjmuje rekord o długości 512(10) bajtów.

FLX przyjmuje następujące tryby przesłania dla zbiorów podanych niżej jako domniemane (z wyjątkiem przesłania zbiorów na taśmie perforowanej)

TRYB	KLUCZ	TYP ZBIORU
obraz zbioru	/IM:n	.TSK,.OLB, .MLB,.SYS,
format binarny	/FB:n	.OBJ,.STB, .BIN,.LDA
format znakowy	/FB:n	wszystkie pozostałe

Jeśli określimy "n" kluczem /FA, /FB lub /IM w przypadku, gdy wyjściowy zbiór nie jest zbiorem o strukturze DOC-PB to, FLX ignoruje "n".

2.3. Klucze sterujące

Poniższa tabela opisuje te klucze.

FLX - Klucze sterujące

KLUCZ	OPIS
/BL:n	Wskazuje liczbę ciągłych bloków (n), które mogą być przydzielone dla zbioru wyjściowego. Klucz ten używamy z kluczem /CO (opisany poniżej). Jeśli klucz /BL nie został określony, wówczas długość zbioru wejściowego jest przyjęta jako długość zbioru wyjściowego. Sposób przydziału zastosowany dla tomów AMKO zazwyczaj przydziela dla nowego zbioru największy dostępny na tomie obszar. Użycie klucza /BL:n z kluczem /RT dla zbioru wyjściowego spowoduje przydzielenie mu pierwszego nieużytego obszaru o długości "n". Jednak gdy, zbiór o strukturze AMKO jest zamykany wówczas rozmiar zbioru wejściowego jest użyty jako rozmiar zbioru wyjściowego. Jeżeli liczba bloków w zbiorze wejściowym nie jest równa "n" wystąpi błąd. Ponieważ wszystkie zbiory o strukturze AMKO są ciągle nie wymaga się klucza /CO z kluczem /BL:n dla zbiorów wyjściowych o strukturze AMKO.
/BS:n	Określa rozmiar bloku (n) w bajtach dla zbiorów wyjściowych na kasetowej taśmie magnetycznej. Jeżeli nie użyjemy klucza /BS, przyjmuje się rozmiar bloku 128 (10). Klucz /BS jest dozwolony dla zbiorów wyjściowych na kasetowej taśmie magnetycznej (CT) określonych kluczem /RS
/CO	Oznacza, że wyjściowy zbiór ma być ciągły. Klucza /CO używamy tylko dla dysków. Jeśli zbiorem wejściowym jest taśma perforowana, kasetowa lub taśma magnetyczna o strukturze DOS-SM konieczne

- jest także użycie klucza /BL. FLX przesyła zbiory typów .TSK,.SYS i .OLB z kluczem /CO do tomów o strukturze DOC-PB, gdy wejście jest tomem o strukturze DOS-SM lub AMKO zapisanym na dysku.
- /DE Usuwa zbiory z dysku o strukturze DOS-SM. Jest także używany z kluczem /RT w celu usunięcia zbiorów z dysku o strukturze AMKO. W linii komendy nie określa się zbiorów wyjściowych.
- /DI Powoduje listowanie katalogu tomów DOS-SM. Jest on tworzony w wyspecyfikowanym zbiorze wyjściowym. Jest używany także z kluczem /RT do utworzenia listingu katalogu dla tomów struktury AMKO w wyspecyfikowanym zbiorze wyjściowym. FLX nie może listować katalogów DOC-PB. Jeśli nie określimy urządzenia wyjściowego to katalog jest wyprowadzany na TI:. Jeżeli nie określimy nazwy i typu zbioru wyjściowego to przyjmuje się konwencje s.s.
- /DNS:n Określa gęstość zapisu na taśmie magnetycznej; n=800 albo 1600. Jeśli n jest jakąkolwiek inną liczbą, albo nie jest określone wówczas FLX drukuje informacje o błędzie. Jeżeli nie użyjemy klucza /DNS:n wówczas domniemana gęstość zapisu na taśmie magnetycznej wynosi 800 bitów na cal. Jeżeli użyjemy klucza /DNS dla urządzenia innego niż taśma magnetyczna, wówczas FLX ignoruje klucz.
- /FC Klucz /FC używamy przy zbiorach typu FORTRAN. Oznacza on, że pewne znaki w zbiorach typu FORTRAN są interpretowane jako znaki sterujące karetką. Klucz /FC stosuje się jedynie do zbioru wyjściowego w strukturze DOC-PB. W podręczniku "FORTRAN - Opis języka" można znaleźć więcej informacji o konwencji

- sterowania karetką w FORTRANie. W podręczniki "Operacje wejścia - wyjścia" zawarto informacje o blokach danych w zbiorze i atrybutach rekordów.
- /ID** Klucz /ID powoduje podanie numeru wersji FLX. Można użyć klucza /ID jako części specyfikacji wyjściowej lub wejściowej zbioru albo użyć go w odpowiedzi na przynaglenie (FLX).
- /LI** Jak klucz /DI.
- /NU:n** Używa się z kluczami /ZE i /RT w celu określenia liczby przydzielonych bloków katalogu (n), podczas inicjacji dysków AMKO. Jeżeli nie użyjemy klucza /NU:n, przydzielone są 4 bloki katalogu. Maksymalna liczba bloków, które mogą być przedzielone wynosi 37(8) (31 dziesiętnie).
- /RW** Przewinięcie taśmy magnetycznej przed rozpoczęciem przesyłania zbiorów. Jeżeli wyspecyfikowany będzie klucz /-RW, FLX rozpoczyna przesłanie zbiorów bez wcześniejszego przewinięcia taśmy magnetycznej. Jeżeli nie jest wyspecyfikowany, przyjmowany jest klucz /RW z domniemania. Jeżeli wyspecyfikowany jest klucz przewinięcia dla urządzenia nie będącego taśmą magnetyczną lub łącznie z kluczem /LI, kluczem /DI czy /ZE, FLX ignoruje ten klucz.
- /SP** Wskazuje, że zbiór wyjściowy będzie wyprowadzony przez systemowy program drukowania lub przez podsystem wyprowadzania. Klucz /SP dotyczy tylko zbiorów wyjściowych typu DOC-PB.
- /UI** Wskazuje, że zbiorom wyjściowym należy nadać takie same UFD jakie mają zbiory wejściowe. FLX ignoruje klucz /UI jeżeli specyfikacja zbioru wyjściowego zawiera UFD. Klucz /UI jest odpowiedni tylko dla zbiorów wyjściowych DOS-SM albo DCC-PB.

- /VE Nakazuje czytać i sprawdzić każdy rekord zapisany na taśmie kasetowej. Klucz /VE może być użyty tylko z kluczem /CT w specyfikacji zbioru wyjściowego.
- /ZE Inicjacja kasetek lub tomów DOS-SM. Może być użyty z kluczem /RT (oraz /NU) do inicjacji tomów AMKO. Inicjacja usuwa wszystkie zbiory uprzednio zapisane na urządzeniu. Klucz /ZE wymaga specyfikacji zbioru.

3. Działanie z taśmą papierową

FLX umożliwia działanie z urządzeniami taśmy papierowej takimi jak czytnik/perforator jako urządzeniami DOS-SM. FLX ma możliwość wydzielania rekordów na taśmie papierowej dla zbiorów o formacie binarnym lub formacie znakowym. FLX specjalnie traktuje zbiory, które zwykle z domniemania są przesyłane bez formatowania tzn. zbiory z rozszerzeniem .TSK, .OLB, .MLB, .SYS, .SML, .ULB. Przy wyprowadzaniu taśmy papierowej zbiór jest zapisany z domniemania w formacie binarnym. Przy czytaniu taśmy papierowej na tomie o strukturze DOC-PB zbiór jest zapisywany z domniemania w 512(10)-bajtowych rekordach stałej długości. Jednak nowe zbiory nie będą ciągle, chyba że w specyfikacji zbioru wyjściowego podano klucz /CO/BL:n. Należy znać odpowiednią wartość n (liczba bloków ciągłych) przed wprowadzeniem komendy. Użycie w sposób jawny kluczy trybu przesłania, podczas przepisywania zbiorów typu: .TSK, .OLB, .MLB, .SYS, .SML i .ULB między taśmą papierową i tomami o strukturze DOC-PB może spowodować, że zbiory powtórnie czytane z taśmy papierowej będą różne od zbiorów, z których były wcześniej przepisywane. Domniemanie to zapewnia, że po powtórным czytaniu ich z taśmy papierowej, będą miały ten sam format, co przed perforacją. Nie można specyfikować nazwy zbioru wyjściowego dla

komend FLX współpracy z taśmą papierową. Nazwa zbioru wprowadzona w specyfikacji zbioru wejściowego jest użyta jako nazwa zbioru wyjściowego. Na przykład:

FLX>DK1:/RS=PR:CRTMAC.DAT/DO

Komenda zapisuje zbiór o nazwie: DK1:CRTMAC.DAT.

Jeżeli nie wyspecyfikuje się nazwy zbioru wejściowego domniemana nazwa zbioru jest: .:n, gdzie "n" określa ostatni numer wersji.

System DOC-PB dopuszcza taśmę papierową tylko jako urządzenie DOS-SM. Należy specyfikować klucz /DO. Przykład poniżej ilustruje specyfikacje taśmy perforowanej dla zbiorów wejściowych i wyjściowych.

FLX>PP:/DO=CRTMAC.DAT/RS

FLX>DK:/RS=PR:CRTMAC.DAT/DO

Przy kopiowaniu zbioru jednej taśmy perforowanej na inną należy używać klucza (/IM) bez względu na format taśmy.

Na przykład:

FLX>PP:/DO/IM=PR:/DO

PROGRAM TESTOWANIA NOŚNIKÓW DYSKOWYCH

1. Wprowadzenie

Program testowania nośników dyskowych BAD testuje nośniki aby określić numery uszkodzonych sektorów. BAD zapisuje na testowanym nośniku informacje o wykrytych uszkodzonych sektorach. W czasie działania komendy INI monitora systemu, sektory te stają się niedostępne, przez przyłączenie ich do zbioru [0,0]BADBLK.SYS. Sektory te nie będą dostępne w innych zbiorach. Urządzenia testowane programem BAD dzielą się na dwie grupy:

- urządzenia typu ostatni sektor. W urządzeniach tych informacja o uszkodzonych sektorach zapisywana jest na ostatnim nieuszkodzonym sektorze
- urządzenia typu ostatnia ścieżka. W urządzeniach tych informacja o uszkodzonych sektorach zapisywana jest na ostatniej ścieżce

Program BAD może być używany w dwóch wersjach: w wersji pracującej pod kontrolą systemu operacyjnego DOC PB oraz w samodzielnej wersji ładowalnej programem ładowania początkowego. Wersja samodzielna jest szczególnie przydatna w przypadku konfiguracji z jedną jednostką dyskową.

UWAGA

Używane w tym opisie określenie "sektor", równoważne jest określeniu "blok fizyczny"

2. Wywołanie programu BAD

Aby wywołać program BAD należy użyć następującą komendę:

```
>BAD ddnn:[/sw]...
```

gdzie:

ddnn: jest specyfikacją urządzenia, składającą się z dwuznakowej nazwy oraz numeru jednostki (oktalnie)

/sw jest opcjonalnym kluczem komendy. Jeżeli komenda składa się z kilku kluczy, to należy je podać w jednym wierszu. Jeżeli żaden klucz nie zostanie podany, to BAD przystępuje do automatycznego testowania poszczególnych sektorów nośnika.

2.1. Klucze programu BAD

Tablica 1 zawiera listę kluczy programu BAD wraz z ich krótkim opisem.

Tablica 1.

Klucze programu BAD

Klucz	Opis klucza
Klucze wspólne dla obu wersji programu	
/LI	listowanie uszkodzonych sektorów po ich wykryciu
/MAN	przejście do manualnego wprowadzania sektorów uszkodzonych
/OVR	polecenie utworzenia opisu sektorów uszkodzonych na urządzeniu typu ostatnia ścieżka, z pominięciem zbiorów na ostatniej ścieżce
/RETRY	zezwoleńie na korygowanie błędów dyskowych
/UPDATE	pozwolenie na poprawianie zbioru opisów sektorów uszkodzonych

Klucze wersji samodzielnej

/CSR=nnnnnn	zmiana adresu rejestru kontrolera
/VEC=nnn	zmiana adresu wektora przerwania
/WCHK	powoduje sprawdzenie poprawności zapisu
/NOWCHK	zapis odbywa się bez sprawdzenia poprawności

2.2. Sprawdzenie nośnika

Bad sprawdza dyski zapisując wzorzec testowy w każdym sektorze na dysku, następnie czyta sek-

tory do bufora w pamięci i porównuje dane z bufora z danymi, które były zapisane w sektorach. Proces testowania nośnika niszczy znajdującą się na nim informację.

Wzorzec testowy pisany jest do kilku sektorów w czasie jednej operacji pisania. Jeśli wystąpi błąd w czasie zapisu, czytania lub porównywania jednego z tych sektorów, wówczas każdy sektor jest sprawdzany indywidualnie. Wzorzec testowy stanowią słowa o wartości 165555 i 133333 (oktalnie) powielone 128. razy w każdym sektorze. Jeśli podsystem rejestracji błędów rejestruje błąd, a program BAD nie wykrywa sektorów uszkodzonych, to powodem tego może być błąd kontrolera dyskowego, ujawniający się czasie długich transmisji danych.

2.3. Współdziałanie komendy INI i programu BAD

BAD wspólnie z komendą INI monitora systemu używany jest do inicjowania nośnika dyskowego i założenia na nim struktury zbiorów systemu DOC PB. Po tych operacjach nośnik staje się tomem dyskowym. Komenda INI używa informacji o uszkodzonych sektorach, utworzonej przez program BAD, do stworzenia zbioru [C,O]BADBLK.SYS. Zbiór BADBLK.SYS zawiera te sektory, które zostały uznane za uszkodzone, zapewniając w ten sposób to, że system zbiorów nie zapisze danych do uszkodzonego sektora.

3. Użycie programu BAD

Przed użyciem programu BAD, dysk musi być sformatowany fabrycznie, lub przez jeden z programów diagnostycznych dostarczanych wraz z systemem. Dysk, który jest weryfikowany nie może być zamontowany, tzn. nie może być wykonana komenda MOUNT dla tego dysku. Jeśli dysk jest zamontowany należy wydać komendę DISMOUNT w celu zdemontowania go.

Niżej przedstawiony przykład ilustruje testowanie nośnika na dysku DK1:

```
>ALL DK1:<CR>  
>FMT DK1:[/sw]<CR>  
>BAD DK1:[/sw]<CR>  
>INI DK1:[etykieta][/sw]<CR>  
>MOU DK1:[etykieta][/sw]<CR>
```

gdzie:

/sw jest kluczem komendy
etykieta jest nazwą tomu dyskowego.

PROGRAM WERYFIKACJI STRUKTURY ZBIORÓW VFY

1. Wprowadzenie do programu VFY

Program weryfikacji struktury zbiorów VFY (File Structure Verification Utility) dostarcza następujących możliwości:

1. Kontrola czytelności i prawidłowości struktury zbiorów tomu o strukturze zbiorów DOC PB (jest to funkcja domniemana, wykonywana bez podawania żadnego klucza).
2. Wyprowadzanie ilości dostępnych bloków w tomie o strukturze zbiorów DOC PB (klucz /FR).
3. Wyszukiwanie zbiorów, które figurują w zbiorze nagłówków, ale nie wchodzi w skład żadnego katalogu w tym tomie (klucz /LO). Zbiory takie nazywane "zgubionymi", w tym sensie, że nie można się do nich odwoływać przez nazwę zbioru.
4. Listowanie wszystkich zbiorów figurujących w zbiorze nagłówków, z podaniem takich informacji, jak identyfikator zbioru ID, nazwa zbioru, właściciel (klucz /LI).
5. Oznaczanie jako zajętych wszystkich takich bloków, które nie są aktualnie zapisane przez zbiór, ale są już objęte alokacją dla zbioru (klucz /UP).
6. Przebudowanie mapy zajętości dysku tak, aby zawierała dokładne odwzorowanie informacji ze zbioru nagłówków (klucz /RE).
7. Przywracanie zbiorów oznaczonych do usunięcia (klucz /DE).
8. Wykonanie odczytu kontrolnego każdego zapisanego bloku w tomie o strukturze zbiorów DOC PB (klucz /RC).

W tomie, na którym działa VFY, nie powinny działać inne zadania. W szczególności nie można w tym tomie tworzyć nowych zbiorów, rozszerzać już istniejących zbiorów, ani też usuwać zbiorów.

VFY nie może być usuwany w czasie wykonywania funkcji /UP, /RE lub /DE.

2. Wiersz komendy VFY

Wykonanie dowolnej funkcji VFY inicjuje się komendą terminala. Format komend VFY jest następujący (każdy z elementów może być pominięty):

>VFY listfile,scratchdev:=indev:/switch
albo

>VFY indev:/switch
(jest to forma skrócona od >VFY TI:,indev:=
indev:/switch)

listfile jest to specyfikacja listingowego zbioru wyjściowego, o postaci:
ddn:[ufd]filename.ext;ver

Brak specyfikacji oznacza TI:, tzn. terminal, z którego wydano komendę.

scratchdev: Określa zbiór, na którym ma powstać zbiór roboczy programu VFY. Brak tego parametru oznacza, że będzie użyte wolne miejsce w tomie na urządzeniu SY:. VFY stosuje zbiór roboczy przy weryfikacji oraz przy wyszukiwaniu zbiorów zgubionych. Po zakończeniu działania VFY zbiór ten jest automatycznie usuwany.

Zaleca się umieszczać zbiór roboczy w innym tomie, niż testowany.

indev: Określa to, który ma być weryfikowany. /switch Określa funkcje, którą VFY ma wykonać. Normalnie VFY działa na weryfikowanym tomie w trybie "wyłącznie do odczytu", przyjmując, że zbiór roboczy, jeżeli wystąpi, zostanie założony na innym urządzeniu. Dostęp do zapisu jest wymagany tylko w następujących przypadkach:

1. Jeżeli użyto klucza /UP lub /RE, to dostęp do zapisu jest wymagany w celu zapisania mapy rozmieszczenia pamięci ([O,O]BITMAP.SYS).
2. Jeżeli określono klucz /DE, to VFY wymaga dostępu do zapisu dla zbioru nagłówek ([O,O]INDEXF.SYS).
3. Jeżeli określono klucz /LO i zostały znalezione zgubione zbiory, to VFY wymaga dostępu do zapisu katalogu UFD=[1,3].

Jeżeli VFY korzysta z dostępu tylko do odczytu, to może być uruchamiany pod dowolnym UIC. Przy dostępie do zapisu może być uruchamiany pod UIC z grup systemowych tj.: UIC=[g,m], gdzie $g \in \{10(8)\}$.

Jeżeli VFY wymaga dostępu do zapisu zbioru nagłówków lub zbioru mapy zajętości, to tom powinien być wcześniej zamontowany z użyciem klucza /UNLOCK.

3. Klucze programu VFY

Klucze programu VFY, zapisywane na końcu wiersza komendy określają funkcje, które VFY ma wykonać. Tablica 1 zawiera zestawienie kluczy i ich skrócone opisy.

3.1. Kontrola poprawności struktury (funkcja domniemana)

Funkcja ta jest podstawową funkcją VFY i polega na sprawdzeniu czytelności i poprawności struktury zbiorów w tomie na określonym urządzeniu. Odbywa się to poprzez odczytanie wszystkich nagłówków zbiorów ze zbioru nagłówków i sprawdzenie dla każdego z nich, czy pole mapy zawarte w nagłówku zbioru wskazuje na bloki, które, jak wynika z mapy zajętości tomu ([O,O]BITMAP.SYS), są w tym tomie przydzielone.

Tablica 1.

Klucze i funkcje VFY

<u>Klucz</u>	<u>Funkcja</u>
Brak klucza	Funkcja domniemana: sprawdza czytelność i poprawność struktury zbiorów zamontowanego tomu
/DE (delete)	Zeruje oznaczniki usunięcia zbiorów
/FR (free)	Wyprowadza informacje o wolnej przestrzeni na dysku

/LI (list)	Listuje cały zbiór nagłówków, z identyfikatorami zbiorów ID
/LO (lost)	Przegląda strukturę zbiorów tomu i wyszukuje zbiory nie znajdujące się w żadnym katalogu
/RC (read check)	Sprawdza poprawność odczytu każdego bloku wszystkich zbiorów tomu
/RE (rebuild)	Odzyskuje bloki, które są wskazane jako przydzielone, a jeszcze nie są zawarte w zbiorach
/UP (update)	Przydziela bloki, które okazują się być dostępne, a zostały przydzielone do zbiorów

Dla wykonania tej funkcji obowiązują następujące wymagania:

1. Weryfikowany tom powinien być zamontowany jako tom o strukturze zbiorów DOC PB.
2. Tom może być zablokowany przed zapisem, ale tylko w następujących sytuacjach:
 - a. gdy nie jest systemowym urządzeniem (SY:) użytkownika
 - b. gdy potrzebny zbiór roboczy zostanie założony w innym tomie o strukturze zbiorów DOC PB

Przykład:

>Vfy DKO:

CONSISTENCY CHECK OF INDEX AND BITMAP ON DKO:

INDEX INDICATES 1145. BLOCKS FREE, 3655.

BLOCKS USED OF 4800.

BITMAP INDICATES 1145. BLOCKS FREE, 3655.

BLOCKS USED OF 4800.

3.1.1. Raportowanie błędów w zbiorach

W trakcie weryfikowania tomu Vfy raportuje na bieżąco wykryte błędy. Każdy z takich komunika-

tów jest poprzedzony wierszem identyfikacji zbioru, który ma następujący format:
FILE ID nn,mm filename.ext;version OWNER [g,m]

3.1.2. Zbiory oznaczone do usunięcia

Jeżeli pewien zbiór został oznaczony do usunięcia, ale operacja usunięcia nie została zakończona, wówczas użytkownik ma zarówno możliwość odzyskania zbioru (gdyby był jeszcze potrzebny), jak również dokończenia operacji usuwania zbioru (aby zwiększyć ilość wolnego miejsca).

Odzyskiwanie zbioru

Aby odzyskać zbiór oznaczony do usunięcia, należy zamontować tom komendą MOU z kluczem /UNL. Na przykład:

```
>MOU DKO:/UNL
```

Następnie uruchamia się VFY z kluczem /DE, co powoduje wyzerowanie wskaźników oznaczenia zbiorów do usunięcia. Kolejnym etapem powinno być uruchomienie VFY z kluczem /LO, w celu zweryfikowania całej struktury zbiorów tomu.

Usuwanie zbioru oznaczonego do usunięcia

Zbiory oznaczone do usunięcia można usuwać bezpośrednio za pomocą programu PIP, gdyż można tu wykorzystać identyfikatory ID tych zbiorów, wprowadzane przez VFY przy weryfikacji tomu. Poniższy przykład ilustruje, jak wykorzystać ID, aby użyć programu PIP do usunięcia zbioru:

```
>PIP /FI:12:20/DE
```

Powyższa komenda spowoduje usunięcie zbioru o ID równym 12,20, z bieżącego dysku. Ukazuje się wprawdzie komunikat:

```
PIP -- FAILED TO MARK FILE FOR DELETE - NO SUCH  
FILE
```

w związku z tym, że system traktuje zbiory oznaczone do usunięcia jako już nieistniejące. Jednakże komenda odnosi pożądaný skutek tzn. zbiór zostaje całkowicie usunięty.

3.1.3. Usuwanie bloków wielokrotnie zaalokowanych

Jeżeli struktura zbiorów zawiera bloki wielokrotnie zaalokowane, należy pousuwać zbiory tak, aby nie było bloków. Automatycznie wykonywany przez VFY przegląd dodatkowy tomu zapewnia wyprowadzenie wykazu takich bloków. Wraz z blokami są także wyprowadzane informacje o zbiorach związanych z takimi blokami. Dzięki tym informacjom użytkownik może określić, które ze zbiorów chce zachować, a pozostałe zbiory można usunąć za pomocą programu PIP.

3.1.4. Eliminacja "wolnych" bloków

Jeżeli w tomie nie ma wielokrotnie zaalokowanych bloków, to kolejnym problemem jest eliminacja bloków, które w mapie zajętości pamięci tomu są oznaczone jako wolne, ale są zaalokowane do pewnego zbioru. Aby spowodować oznaczenie tych bloków jako zajętych w ogólnej mapie zajętości, należy użyć VFY z kluczem /UP. Wówczas zostaną one zaalokowane do zbiorów.

3.1.5. Przywracanie bloków zgubionych

Aby określić, czy w tomie o strukturze zbiorów DOC PB są jakieś bloki zgubione, należy sprawdzić treść ostatnich dwóch wierszy raportu, jaki daje VFY przy kontroli poprawności. W wierszach tych unajduje się informacja o wolnej przestrzeni w tomie. W pierwszym z tych wierszy jest wypisana ilość dostępnego miejsca zgodnie z zapisami w zbiorze nagłówków, tzn. ile bloków tomu nie jest objętych zbiorami zarejestrowanymi w zbiorze nagłówków. W drugim wierszu jest wypisana informacja o ilości wolnego miejsca wynikająca z mapy zajętości pamięci tomu.

Jeżeli założymy, że tom nie ma usterek, to te dwie informacje powinny być ze sobą zgodne. Natomiast różnica pomiędzy drugą i pierwszą z

nich jest to ilość bloków zgubionych, tzn. takich, które są wykazywane jako zajęte, ale nie należą do żadnego zbioru.

Bloki zgubione można przywrócić, wykonując VFY z kluczem /RE.

3.2. Klucz /DE (DELETE)

Klucz /DE powoduje wyzerowanie wskaźników oznaczenia zbiorów do usunięcia, które to wskaźniki znajdują się w obszarze nagłówka zbiorów i dotyczą zbiorów oznaczonych do usunięcia, ale jeszcze dotąd nie usuniętych.

Tom musi być wcześniej zamontowany z kluczem /UNL.

3.3. Klucz /FR (FREE)

W wyniku działania klucza /FR otrzymuje się następujący komunikat:

ddn: HAS nnnn. BLOCKS FREE, nnn. BLOCKS USED
OUT OF nnnn.

(Urządzenie ddn: ma nnnn. wolnych bloków, nnnn. bloków zajętych z ogólnej liczby nnnn. .)

3.4. Klucz /LI (LIST)

Działanie klucza /LI polega na wylistowaniu całego zbioru nagłówków, w kolejności identyfikatorów zbiorów ID. Dla każdego ze zbiorów jest wprowadzany numer aktualny zbioru, numer kolejny zbioru, nazwa zbioru, konto właściciela.

VFY)DK:/LI

LISTING OF INDEX ON DKO:

```
FILE ID 000001,000001 INDEXP.SYS;1 OWNER [1,1]
FILE ID 000002,000002 BITMAP.SYS;1 OWNER [1,1]
FILE ID 000003,000003 BADBLK.SYS;1 OWNER [1,1]
FILE ID 000004,000004 000000.DIR;1 OWNER [1,1]
FILE ID 000005,000005 CORIMG.SYS;1 OWNER [1,1]
FILE ID 000006,000006 001001.DIR;1 OWNER [1,1]
FILE ID 000007,000007 001002.DIR;1 OWNER [1,2]
FILE ID 000010,000010 EXEMC.MBL;1 OWNER [1,1]
```

FILE ID 000012,000012 NODES.TBL;1 OWNER [1,1]
FILE ID 000013,000036 QIOSYM.MSG;311 OWNER[1,2]
FILE ID 000014,000037 F4PCOM.MSG;1 OWNER [1,2]

Rys. 1 Przykładowy listing programu VFY z kluczem /LI

3.5. Klucz /LO (LOST)

Klucz /LO powoduje, że VFY przegląda strukturę zbiorów tomu i wyszukuje zbiory zgubione, tzn. zbiory nie należące do żadnego katalogu. Zbiory zgubione nie są dostępne przez nazwę zbioru. VFY podaje listę zgubionych zbiorów oraz włącza je do katalogu [1,3] ("katalogu zgubionych zbiorów"), o ile katalog ten został wcześniej założony w danym tomie.

3.6. Klucz /RC (READ CHECK)

Użycie klucza /RC powoduje sprawdzenie poprawności odczytu każdego bloku wszystkich zbiorów danego tomu.

W związku z tym, że funkcja /RC wymaga dostępu tylko do odczytu, dany tom może być sprzętowo zabezpieczony przed zapisem.

Obszar buforowy VFY może być większy lub mniejszy, zależnie od wielkości partycji, w której VFY jest zainstalowany.

W celu maksymalnego przyspieszenia weryfikacji należy stosować możliwie największy współczynnik blokowania. Za każdym razem, gdy jest wydrukowany błąd odczytu, kolejne bloki danej porcji są odczytywane indywidualnie, aby określić, który blok /bloki/ danych nie może być odczytany.

Po wykryciu błędu listowany jest wiersz identyfikacyjny zbioru, w następującym formacie:

```
FILE ID nn,nn filename.ext;ver. blocks used/  
      blocks allocated
```

gdzie

blocks used - oznacza ilość bloków wykorzystanych przez zbiór,

blocks allocated - oznacza ilość bloków przydzielonych dla zbioru.

Następnie jest wypisywany komunikat błędu. Jeżeli współczynnik jest inny niż 1, to ten komunikat ma następujący format:

ERROR STARTING AT VBN n1,n2 LBN n1,n2 - ERROR CODE -n

gdzie

VBN - oznacza wirtualny numer bloku,

LBN - oznacza logiczny numer bloku.

Przy współczynniku blokowania równym jedności powyższy komunikat nie ukazuje się. Niezależnie od tego mogą wystąpić komunikaty błędów odnoszące się do poszczególnych bloków porcji, tzn. do tych bloków, w których występują błędy odczytu.

Są to komunikaty o następującym formacie:

ERROR AT VBN n1,n2 LBN n1,n2 - ERROR CODE -n

Jeżeli komunikat ERROR STARTING AT ukazał się bez następujących po nim (jednego lub więcej) komunikatów ERROR AT, wówczas oznacza to, że operacja odczytu wieloblokowego wykonuje się z błędem, natomiast poszczególne bloki porcji dają się odczytać poprawnie.

3.7. Klucz /RE (REBUILD)

Klucz /RE przywraca bloki zgubione tzn. bloki wykazywane jako zajęte, a nie należące do żadnego zbioru.

Przed użyciem tej funkcji należy usunąć ze zbioru bloki wielokrotnie zaalokowane.

Tom, dla którego stosuje się /RE, musi być dostępny do zapisu.

VFY z kluczem /RE uruchamia się pod UIC z grupy użytkowników systemowych.

3.8. Klucz /UP (UPDATE)

Klucz /UP powoduje przydzielenie tych bloków, które w mapie zajętości tomu są wykazywane jako wolne, a tymczasem należą do pewnych zbiorów w tomie, co wynika z map w nagłówkach zbiorów.

Przed zastosowaniem tej funkcji należy usunąć z tomu zbiory z blokami wielokrotnie zaalokowanymi.

Tom, dla którego stosuje się /UP, musi być dostępny do zapisu.

· VFY z kluczem /UP uruchamia się pod UIC z grupy użytkowników systemowych.

PODSYSTEM WYPROWADZANIA PRT& QMG

1. Komenda PRINT i zarządzanie kolejką - wprowadzenie

Komenda Print powoduje, że wydruki zadań użytkowych zapisane na dysku są umieszczane w kolejce, którą steruje Program Zarządzania Kolejką QMG (Queue Manager). Z kolejki tej wydruki są przekazywane w odpowiedniej kolejności do programów zajmujących się ich wyprowadzaniem.

SPOOLNG włączanie wydruków do kolejki - jest to zespół czynności wstępnych wykonywanych przez Podsystem Wyprowadzania dla danego zlecenia drukowania: przejęcie wydruków od zadania, umieszczenie go w urządzeniu pamięci masowej, wprowadzenie do kolejki.

SPOOLER zadanie włączania wydruków do kolejki - zadanie polegające na zorganizowaniu włączenia danego wydruku do Podsystemu Wyprowadzania, a więc na wykonaniu czynności wyżej wymienionych.

DESPOOLER zadanie wyprowadzania systemowego - zadanie stanowiące końcowy etap działania Podsystemu Wyprowadzania dla danego zlecenia drukowania. Wydruk jest w zorganizowany sposób przekazywany z urządzenia pamięci masowej na urządzenie wyjściowe.

Komenda Print wymaga obecności w systemie Programu Zarządzania Kolejką QMG (Queue Manager). Nie we wszystkich instalacjach jest on obecny.

>PRI spec-zbioru

Komenda Print umieszcza nową pozycję w zbiorze SY:[1,7]QUEUE.SYS. Pozycja ta ma unikalny numer i nazwę opartą na pierwszych sześciu znakach nazwy zbioru. Jest umieszczana przez domniemanie w kolejce PRINT. Jeżeli drukarka jest dostępna, to zbiór jest przekazywany z Podsysteme-

mu Wyprowadzania na drukarkę i oddrukowany, po czym znika z kolejki QUEUE.SYS. Zawartość kolejki może być wyświetlana w całości lub częściowo przez użycie odpowiednich komend QUE, realizowanych przez Program Zarządzania Kolejką QMG.

Komenda Print dopuszcza wielokrotne specyfikacje zbiorów. W takim przypadku zbiory występują kolejno po sobie na tym samym urządzeniu wyjściowym, w tej kolejności, w jakiej podano ich specyfikacje. Komenda Print dopuszcza także konwencję gwiazdki.

Jedna komenda Print definiuje jedno zlecenie drukowania dla Programu Zarządzania Kolejką QMG. Zlecenie drukowania może obejmować jeden lub więcej zbiorów.

Można określić szereg atrybutów definiowanego zlecenia drukowania, poprzez użycie kluczy komendy Print. Mogą to być następujące atrybuty:

- Czas, w którym drukowanie ma być wykonane
- Urządzenie, które ma przyjąć wydruk
- Priorytet zlecenia w kolejce
- Restartowalność zlecenia drukowania
- Formaty drukowania
- Ilość wierszy na stronie
- Ilość kopii każdego zbioru, jaka ma być wydrukowana
- Określenie, czy po wyprowadzeniu zbioru ma być usunięty

Komendy programu QMG pozwalają użytkownikowi wyświetlać kolejki lub poszczególne zlecenia, a także modyfikować większość atrybutów zleceń. Jest możliwość wstrzymania zlecenia w kolejce, zwolnienia go po wstrzymaniu, usunięcia zlecenia z kolejki.

Jak już wcześniej powiedziano, komenda Print umieszcza zlecenia drukowania w kolejce. Do umieszczenia w kolejce zlecenia drukowania można wykorzystać również Program Manipulacji Zbiorami PIP. Np. komenda:

```
PIP> NAZWA.TYP /SP
```

powoduje, że określony zbiór znajdzie się w kolejce. Nazwa zlecenia drukowania zostanie utworzona na podstawie nazwy zbioru.

2. Komenda PRINT

Format komendy PRINT jest następujący:

```
>PRINT [[ddnn:] jobname[/jsw]= filespec[/fsw]
      [filespec[/fsw]]...
```

Komenda PRINT przekazuje zlecenie drukowania do Programu Zarządzania Kolejką QMG, w celu systemowego wyprowadzenia.

ddnn: Określa urządzenie wyprowadzania systemowego dla danego zlecenia drukowania. Każde urządzenie wyjściowe obsługuje oddzielną, właściwą dla niego kolejkę. Kolejki mają nazwy pochodzące od nazw urządzeń, a mianowicie: LPQO jest nazwą kolejki związanej z drukarką LPO:, TTQ21 jest nazwą kolejki związanej z terminalem TT21: /jeżeli ustalono ten terminal jako systemowe urządzenie wyprowadzające/ itp. Zatem, aby spowodować przekazanie zlecenia drukowania do kolejki określonego urządzenia, należy umieścić odpowiedni element ddnn: w wierszu komendy PRINT. Jeżeli zbiór określonej kolejki nie istnieje, wówczas cała komenda jest ignorowana.

Urządzenie ddnn: musi być wcześniej określone jako urządzenie wyprowadzania systemowego.

Pominięcie elementu ddnn: spowoduje przesłanie zlecenia do domniemanej kolejki PRINT.

jobname Nazwa zlecenia, składa się 1-6 znaków alfanumerycznych. Pod tą nazwą zlecenie drukowania jest znane w kolejce. Występuje ona również na stronie wignetowej /nagłówkowej/ zlecenia.

W przypadku pominięcia elementu jobname nazwa zlecenia jest tworzona z 6 początkowych znaków nazwy pierwszego zbioru. Jeżeli dla pierwszej nazwy zbioru zastosowano konwencję gwiazdki, to nazwa zlecenia drukowania pochodzi od pierwszego zbioru.

Każde zlecenie ma także przypisany numer miejsca w kolejce, który jest widoczny w wykazie uzyskiwanym komendą QUEUE /LI.

/jsw Określa klucze związane ze zleceniem. Umieszcza się je na lewo od znaku równości (=), zaś sam znak równości jest wymagany w przypadku użycia jednego lub więcej kluczy. Określenia kluczy mogą być skracane do dwóch znaków.

Istnieje 7 kluczy tej grupy. Są one poniżej opisane. Ustawienie każdego z nich, oprócz HOLD, może być zmieniane komendą QUE /MOD.

1. /AFTER:hh:mm

Zlecenie będzie czasowo zablokowane w kolejce do momentu określonego przez :hh:mm. Wartości hh mogą być z zakresu 0-23, zaś mm z zakresu 0-59. Jeżeli klucz ten nie jest zastosowany, to zlecenie od razu wchodzi do kolejki /od godziny 0:00 dnia bieżącego/. Nadejście określonego momentu czasu powoduje zwolnienie zlecenia w kolejce. Jeżeli w tym momencie w kolejce przed tym zleceniem znajdują się inne zlecenia, to zwalniane zlecenie będzie musiało poczekać, aż zostaną one zrealizowane.

2. /INQ]Flag

Zastosowanie klucza /FL powoduje, że oprócz jednej lub kilku stron winietowych zlecenia poprzedzających wydruk całego zlecenia, pojawia się jedna lub kilka stron winietowych

dla każdego zbioru wchodzącego w skład zlecenia wydruku.

Generalnym domniemaniem dla zbiorów jest /NOFLAG. Są wyprowadzane tylko strony winietowe zlecenia, natomiast zbiory są wyprowadzone sekwencyjnie i rozdzielane tylko przez znak wysuwu strony <FF>.

Opcja SYSGEN umożliwia przesłonięcie tego domniemania i określenie stałego wyprowadzania stron winietowych dla zbioru. Z kolei, jeżeli ta opcja SYSGEN była zastosowana, to użycie klucza /NOFL przesłania działanie tej opcji i strony winietowe zbiorów nie będą wyprowadzane.

Strona winietowa zlecenia zawiera m.in. bieżący UIC użytkownika, który wydał komendę PRINT, a także nazwę zlecenia drukowania - obie te informacje są wyprowadzone w dużym ozdobnym formacie.

Strona winietowa zbioru zawiera m.in. nazwę, typ i numer wersji zbioru, wypisane w dużym ozdobnym formacie.

3. /Priority:n

Ten klucz ustawia priorytet zlecenia w kolejce. Wartość n podaje się z zakresu 0-250. Wartość domniemana n=50. Program Zarządzania Kolejką umieszcza zlecenie w kolejce zgodnie z jego priorytetem. Zlecenia o takim samym priorytecie są obsługiwane w kolejności nadejścia.

4. /HOLD

Klucz ten powoduje wstrzymanie zlecenia w kolejce aż do momentu zwolnienia go komendą QUE jobname /RElease. Zastosowanie /HOLD zilustrowano przykładami podanymi na końcu tego punktu.

5. /LEnsth:n

Klucz ustala ilość wierszy na stronie wydruku. Liczbę n podaje się z zakresu 1-65535.

Zastosowanie tego klucza powoduje wstawie - nie znaku wysuwu strony <FF>, jeżeli nie wystąpił on w drukowanym zbiorze na przestrzeni ostatnich n wierszy. Przez domniemanie przyjmuje się nieokreśloną długość strony.

6. /FOrM:n

Klucz te ustala ustawienie formatu strony, jaki powinno mieć urządzenie, aby mogło zrealizować dane zlecenie drukowania. Liczby n przypisane do poszczególnych formatów posiadanych przez urządzenia instalacji stanowią informację systemową ustaloną dla danego systemu - zatem można ją uzyskać od programisty systemowego. Przy zastosowaniu tego klucza zlecenie czeka, aż będzie dostępne urządzenie wyjściowe o zadanych cechach. Wartością domniemaną n jest 0, co zwykle oznacza ustawienie standardowe.

7. /[NO]REstart

Ten klucz powoduje restartowalność /nierestartowalność/ zlecenia drukowania. Jeżeli użyje się klucza /REstart i realizacja zlecenia drukowania zostanie z jakiegoś powodu przerwana /np. upadek systemu/, to wydruk jest powtarzany od początku pierwszego zbioru zlecenia. Wartością domniemaną klucza jest /NOREstart. Przy tym ustawieniu po przerwaniu zlecenia będzie ono wznowione od początku zbioru, w którym nastąpiło przerwanie drukowania.

filespec Określa zbiór do wydrukowania. Użytkownicy mogą drukować dowolne zbiory, o ile mają do nich dostęp do odczytu (R). Odnośnie elementów specyfikacji zbioru mają zastosowanie zwykle reguły domniemań i konwencja gwiazdki. Domniemanym typem zbioru jest .LST. W przypadku kilku specyfikacji zbiorów oddziela się je przecinkami.

Od momentu umieszczenia zlecenia w kolejce do momentu wydruku zbiory objęte zleceniem wydruku nie powinny być przemianowane ani usunięte. Również tom zawierający zbiór lub zbiory objęte zleceniem drukowania nie może być zdemontowany ani fizycznie zdjęty w czasie przebywania zlecenia w kolejce. W przypadku zdemontowania tomu zlecenie pozostaje w kolejce, drukowane są strony winietowe zlecenia, a następnie drukowany jest komunikat o błędzie otwarcia zbioru.

/fsw Oznacza klucze związane ze zbiorami. Mogą one wystąpić po dowolnej specyfikacji zbioru w wierszu komendy. Zastosowanie klucza dla pewnej specyfikacji zbioru rozciąga się również na dalsze specyfikacje, chyba, że zostanie przesłonięte przez użycie innej wartości klucza. Podano na to przykłady w końcu tego punktu. Istnieją 2 klucze tej grupy, opisane poniżej.

1. /COPIES:n
Klucz ten ustala ilość kopii zbioru, które mają być sporządzone. Kopie te są drukowane kolejno po sobie, a dopiero potem następuje wydruk tej samej /bądź innej, jeśli zostanie zmieniona/ ilości kopii następnego zbioru. Wartością domniemaną n jest 1.
2. /NODELETE
Użycie tego klucza nakazuje Programowi Zarządzania Koleją usunięcie zbioru z katalogu po jego wyprowadzeniu. Wartością domniemaną klucza jest /NODELETE.

Przykłady

>PRINT JAM

Ostatnia wersja zbioru JAM.LST z domniemanego urzędnika i katalogu będzie przekazana do dom-

niemanej kolejki PRINT, z przyjęciem wszystkich innych domniemań.

PRI /FL=PAUL.MAC,ALAN.MAC /CO:2,TOM.MAC
Komenda PRINT określa zlecenie drukowania złożone z jednej kopii zbioru PAUL.MAC, dwóch kopii zbioru ALAN.MAC i dwóch kopii zbioru TOM.MAC. Każda kopia każdego zbioru będzie poprzeczona przez strony winietowe zbioru.

>PRI /FORM:2 /PRIORITY:75=PAYROLL
Nazwą zlecenia jest PAYROLL. Zlecenie ma priorytet 75, ale nie będzie zrealizowane, dopóki nie będzie dostępne urządzenie o odpowiednim formacie wydruku (FORM:2) - np. urządzenie do drukowania rachunków. Nazwą zbioru jest PAYROLL.LST.

3. Nieuprzywilejowane komendy Programu Zarządzania Kolejką

Format nieuprzywilejowanych komend Programu Zarządzania Kolejką jest następujący:

>QUE job-id /switch[:option]

Wszystkie komendy Programu Zarządzania Kolejką zapisuje się w jednym wierszu /bez kontynuacji/.

Pole identyfikatora zlecenia job-id

Identyfikator zlecenia job-id może być wyrażony zarówno jako nazwa zlecenia nadana w wierszu komendy, jak również jako numer pozycji w kolejce przypisany do zlecenia drukowania przez Program Zarządzania Kolejką. Numer ten jest unikalny i występuje w raporcie uzyskiwanym komendą QUE /LI. Poniżej następuje szczegółowy opis tych dwóch sposobów wyrażenia job-id.

[ddnn:[uic]]jobname

Nazwa zlecenia (jobname) stanowi wystarczający identyfikator zlecenia w przypadku, gdy zlecenie drukowania pochodzi z tego samego konta UIC, co UIC zgłoszenia użytkownika do systemu

oraz, gdy w komendzie PRINT definiującej to zlecenie nie określono urządzenia. Przez symbol uic w powyższym zapisie rozumie się [g,m], zatem zapis [uic] oznacza, że całe określenie UIC jest opcjonalne.

Jeżeli komenda ma objąć tylko zlecenia z kolejki określonego urządzenia, to umieszczamy element ddnn: w polu job-id. (Każde z urządzeń wyjściowych ma bowiem oddzielną kolejkę, której nazwa jest związana z nazwą urządzenia).

/JOB:n1:n2

Tutaj format numeru pozycji w kolejce :n1:n2 jest inny niż w zestawieniu uzyskiwanym komendą QUE /LI, gdzie ma format (n1,n2). Takie określenie job-id służy do zidentyfikowania pojedynczego zlecenia drukowania, o unikalnym numerze pozycji w kolejce.

Komendy w opisywanej postaci pozwalają nieuprzywilejowanemu użytkownikowi realizować następujące funkcje:

- Listowanie pozycji kolejki, w różnych formatach
- Modyfikacja atrybutów zleceń z komend wcześniej wydanych
- Wstrzymywanie zleceń w kolejce
- Zwalnianie zleceń wcześniej wstrzymanych
- Usuwanie zleceń z kolejki

3.1. Komenda QUEUE /LIST

Komendy tej używa się do listowania pozycji kolejki. Jeżeli nie podaje się opcji ani nazwy urządzenia wyjściowego, to jest wyświetlana domniemana kolejka PRINT.

Po wydaniu tej komendy jest wyświetlana również informacja o przypisaniu kolejki do urządzeń wyjściowych.

Format

>QUE job-id /sw[:option]

<u>Klucze</u>	<u>Opcje</u>
/BRIEF	:QUE
/LIST	:ALLQUE
/FULL	:DEV
	:ALLDEV

Określenie identyfikatora job-id jest tu możliwe sposobami wyżej opisanymi, ale jest tu jeszcze trzeci sposób:

[uic]

Taka postać identyfikacji zlecenia powoduje wyświetlenie z kolejek wszystkich zleceń, pochodzących z danego UIC.

Klucze

/BRIEF

Klucz ten powoduje, że zlecenia będą wyświetlane w formacie skróconym. Format skrócony obejmuje dla każdego zlecenia następujące informacje.

1. Nazwa zlecenia
2. UIC, z którego wydano komendę PRINT
3. Numer pozycji w kolejce w postaci (n1,n2)
4. Stan zlecenia - jest to jeden z następujących stanów:
 - Zlecenia aktywne - ACTIVE
 - Zlecenie wstrzymane - HELD
 - Zlecenie czasowo zablokowane - PRINT AFTER
 - Zlecenie w stanie oczekiwania - te stan oznacza się spacjami w miejscu przeznaczonego na oznaczenie stanu zlecenia

/LIST

Format standardowy zestawienia QUE /li obejmuje dla każdego ze zleceń następujące informacje: nazwa zlecenia; UIC, z którego wydano komendę PRINT; numer pozycji w kolejce; stan zlecenia; specyfikacje wszystkich zbiorów wchodzących w skład zlecenia.

W przypadku, gdy w komendzie PRINT nie określono restartowalności zlecenia, wymieniony wykaz specyfikacji zbiorów nie obejmuje zbiorów już wyprowadzonych, lecz tylko zbiorów będących w trakcie wyprowadzania oraz pozostałe zbiory. W przypadku określenia restartowalności wykaz ten obejmuje wszystkie zbiory, których dotyczy zlecenie.

/FULL

Pełny format zestawienia obejmuje dla każdego ze zleceń następujące informacje:

1. Nazwa zlecenia
2. UIC, z którego wydano komendę PRINT
3. Numer pozycji zlecenia w kolejce (n1,n2)
4. Stan zlecenia
5. Wykaz specyfikacji zbiorów niezbędnych do zrealizowania zlecenia
6. Wykaz atrybutów zlecenia, zarówno ustalonych jawnie, jak i przyjętych przez domniemanie

Opcje

- :QUE** Jest to domniemana wartość opcji. Powoduje wyświetlenie pozycji z domniemanej kolejki PRINT.
- :ALLQUE** Powoduje wyświetlenie wszystkich pozycji kolejek. :DEV powoduje wyświetlenie charakterystyki urządzenia. Jeżeli urządzenie nie jest określone w polu job-id, to wystąpi komunikat o błędzie.
- :ALLDEV** Powoduje wyświetlenie charakterystyk wszystkich urządzeń wyprowadzania systemowego.

Przykłady

```
>QUE PRTJOB /LIST /FU
```

```
** PRINT QUEUES **
```

```
PRINT => (LP1:)
```

```
[ 301,370] PRTJOB (2140,162) FORM:0
```


PRI:50 LEN:0 PAGE:0 NORESTART NOFLAG
DKO:[301,370]PRTJOB.TXT;22 COP:1
NODELETE

W tym przykładzie powstaje zestawienie dla zlecenia PRTJOB z domniemanej kolejki PRINT, w pełnym formacie. Zawiera ono m.in. pełny wykaz atrybutów zlecenia, zarówno podanych jawnie, jak i pochodzących z domniemania.

Nawiasy okrągłe przy nazwie urządzenia (LP1:) w tym przykładzie oznaczają, że urządzenie zostało zatrzymane przez Program Zarządzania Kolejką, na zlecenie operatora.

>QUE /LI:ALL

PRINT QUEUES

PRINT => LPO:

[7,372] RELABE (2100,130) ACTIVE ON LPO:

DKO:[7,372]RELABEL.FLO;2

[7,372] LPDRV (2000,134) HELD

DKO:[7,372]LPDRV.MAC;2

LPQO => LPO:

[222,333] CORFIL (2200,136) PRINT AFTER
18-DEC-82 17:00

DKO:[222,333]CORFIL.COR;4

Tutaj są wyświetlone wszystkie zlecenia ze wszystkich kolejek, w formacie standardowym. Są wyświetlane wszystkie zlecenia, bo nie wystąpiło pole job-id. Są wyświetlane wszystkie kolejki, bo użyto opcji :ALL.

3.2. Komenda QUEUE /MOD

Komenda QUE /MOD służy do modyfikacji atrybutów zlecenia przypisanych mu przez komendę PRINT. Mogą być modyfikowane tylko atrybuty niżej wymienione.

Format

>QUE job-id /MODify /sw

Klucze

/AFTER:hh:mm

/NOJFLAG

/LENsth:n

/Priority:n
/Form:n
/[NO]Rstart

Klucze

W tej komendzie klucze służą do wskazania atrybutów zlecenia, które mają być modyfikowane. Jak oznaczono w poniższych formatach kluczy, ich zapisy mogą być skracane do dwóch liter. Znaczenie poszczególnych atrybutów opisano w punkcie poświęconym komendzie PRINT, ustalającej ich wartości. Bieżące wartości atrybutów można wyświetlić za pomocą komendy QUE /FU.

/After:hh:mm

Modyfikuje określenie czasu realizacji danego zlecenia.

/[NO]Flag

Modyfikuje atrybut zlecenia, określający drukowanie/pominięcie stron winietowych.

/Length:n

Modyfikuje atrybut LENGTH zlecenia, określający ilość wierszy na stronie.

/Priority:n

Modyfikuje priorytet zlecenia w kolejce.

/Form:n

Modyfikuje atrybut FORM zlecenia.

/[NO]Rstart

Modyfikuje atrybut RESTART zlecenia.

3.3. Komenda QUEUE /HOLD

Komenda ta służy do wstrzymania zlecenia w jego kolejce. Do momentu zwolnienia wstrzymane zlecenie nie bierze udziału we współzawodnictwie zleceń kolejki, nie może być więc zrealizowane.

Format

>QUE job-id /HOLD

Zlecenie będące w stanie ACTIVE, czyli zlecenie aktualnie wykonywane, nie może być wstrzymane kluczem /HOLD.

Przykład

>QUE BIGJOB /HOLD

Zlecenie drukowane BIGJOB z bieżącego urządzenia i bieżącego UIC zostaje wstrzymane w kolejce PRINT.

3.4. Komenda QUEUE /RELEASE

Komenda ta służy do zwolnienia zlecenia, które wcześniej było wstrzymane w kolejce przez działanie klucza /HOLD, a więc do zwolnienia zlecenia będącego w stanie HELD. Nie można tą komendą zwolnić zlecenia zablokowanego czasowo tj. będącego w stanie PRINT AFTER. W takim przypadku można natomiast użyć komendy QUE /MOD /AFTER, aby zwolnić takie zlecenie.

Format

>QUE job-id /RELease

Przykłady

>QUE READY /REL

Zlecenie o nazwie READY z domniemanego urządzenia i z domniemanego UIC zostaje zwolnione i będzie przetwarzane.

>QUE LATER /REL

>QUE -- JOB DOES NOT EXIST

Użytkownik próbował zwolnić zlecenie będące w stanie PRINT AFTER tj. zlecenie czasowo zablokowane. Być może, zlecenie LATER istnieje, ale nie jest to żadne ze zleceń o stanie HELD.

3.5. Komenda QUEUE /DELETE

Komenda ta służy do usuwania zlecenia drukowania z kolejki. Nie ma związku pomiędzy tą komendą a kluczem /DE programu PIP czy kluczem /DE komendy PRINT. Komenda QUE /DELETE nie usuwa bowiem zbioru z katalogu w tomie.

Format

>QUE job-id /DELETE

4. Instalowanie Podsystemu Wyprowadzania

W celu wygodnego zarządzania wielostrumieniowym wyprowadzeniem na drukarki, w większości systemów stosuje się Podsystem Wyprowadzania, a w nim - Program Zarządzania Kolejką QMG (Queue Manager). W mniejszych systemach można stosować Szeregowy Program Wyprowadzający PRT... (Serial Despooler).

Podsystem Wyprowadzania zawiera w sobie zadanie QMGPRRT, które jest instalowane jako PRT..., ale nie jest to Szeregowy Program Wyprowadzający, lecz program, który służy do przechwytywania do Podsystemu wydruków z kluczem PIP /SP oraz wydruków innych zadań pomocniczych i systemowych. Na Podsystem Wyprowadzania składają się następujące części:

1. Interpreter Wiersza Komendy CLI, który przetwarza zarówno zlecenia drukowania (PRI), jak i komendy dostępu do kolejki (QUE). CLI komunikuje się z Programem Zarządzania Kolejką QMG przez przesyłanie pakietów danych opisujących zlecenia drukowania.
2. Program Zarządzania Kolejką QMG, który steruje umieszczaniem zleceń drukowania w kolejkach, może usuwać zlecenia. QMG komunikuje się z CLI oraz z zadaniem PRT... .
3. Prototypowe Zadanie Wyprowadzania Systemowego LPP, zwane także Zadaniem Wyprowadzania Systemowego. Zadanie to przywiązuje do siebie urządzenie i wyprowadza na nie wydruki zleceń drukowania. Zadanie to należy zainstalować oddzielnie dla każdego urządzenia wyprowadzania systemowego. Instaluje się je pod nazwami ddPn, gdzie dd jest symbolem urządzenia, zaś n - numerem jednostki. Urządzenia wyprowadzania systemowego pozostają pod kontrolą Podsystemu Wyprowadzania.

CLI przyjmuje wiersz komendy np. PRINT i interpretuje zlecenie drukowania w nim zawarte.

Następnie przesyła te informacje do QMG, który tworzy zlecenie drukowania stanowiące jedną pozycję w kolejce. Opis tego zlecenia zawiera specyfikacje zbiorów do drukowania oraz atrybuty zlecenia. Kolejka /kolejki/ jest zapamiętana w zbiorze dyskowym SYO:[1,7]QUEUE.SYS. Kolejki są przypisane do poszczególnych urządzeń wyprowadzania systemowego, które są pod kontrolą zleceń wyprowadzania. Kilka kolejek może być przypisanych do tego samego urządzenia. W momencie, gdy jakieś urządzenie kończy realizację zlecenia drukowania, QMG wybiera następne zlecenie z kolejki /lub kolejek/ przypisanych do tego urządzenia. Po zrealizowaniu wydruku odpowiednia pozycja kolejki jest usuwana. W dalszej części tego punktu zawarte jest opisowe wprowadzenie do procedury wdrożenia Podsystemu Wyprowadzania w danej instalacji, szczegółowe opisy uprzywilejowanych komend Programu Zarządzania Kolejką WMG, a także listing przykładowego zbioru komend, który może być wpisany do systemowej procedury STARTUP.

4.1. Opisowe wprowadzenie i przegląd

W tym punkcie opisano, jak uruchomić Podsystem Wyprowadzania. Czynność tą można wykonać w trybie interakcyjnym, wydając kolejne komendy. Wygodniej jest jednak umieścić odpowiedni zestaw komend w zbiorze STARTUP.CMD, przez co będą one wykonywane automatycznie przy każdym ładowaniu systemu.

Podany niżej sposób postępowania należy zaadoptować do konkretnego systemu. Niżej zamieszczone komendy QUE uruchomienia Podsystemu dopuszczają pewną elastyczność. Jest też możliwa dynamiczna modyfikacja Podsystemu Wyprowadzania. A oto czynności prowadzące do uruchomienia Podsystemu Wyprowadzania.

1. Zainstalować QMG - Program Zarządzania Kolejką. Jest to zadanie uprzywilejowane.

```
>INS QMG /PRI=70
```

2. Zainstalować LPP - Zadanie Wyprowadzania Systemowego.

```
>INS LPP /PRI=70 /TASK=ddPn
```

dd oznacza nazwę urządzenia, zaś n numer jednostki. Zadanie LPP należy zainstalować oddzielnie dla każdego fizycznego urządzenia, które ma być urządzeniem wyprowadzania systemowego. Każde z zadań przywiązuje sobie określone urządzenie. Może być do 15 takich urządzeń.

Pominięcie klucza /TASK oznacza, że LPP będzie zainstalowane pod nazwą LPPO i związane z drukarką LPO:.

3. Zainstalować PRT...

```
>INS QMGPRT
```

Zadanie to instaluje się automatycznie pod nazwą PRT... . Służy ono jako połączenie pomiędzy Podsystemem Wyprowadzania a programem PIP i innymi zadaniami, które wykorzystują PRINTX lub .PRINT. Dzięki niemu Podsystem może akceptować np. zlecenia drukowania wynikiem z użycia klucza PIP /SP.

Podana komenda INS QMGPRT spowoduje automatyczne usunięcie (/REM) Szeregowego Programu Wyprowadzającego PRT... , jeśli by on zainstalowany.

4. Zainstalować CLI - jako Interpreter Wiersza Komendy PRI i oddzielnie jako Interpreter Wiersza Komendy QUE.

```
>INS QMGCLI /TASK=...PRI
```

```
>INS WMGCLI /TASK=...QUE
```

CLI obsługuje zarówno komendy PRI, jak i QUE, dlatego musi być dwukrotnie zainstalowany.

5. Uruchomić QMG - Program Zarządzania Kolejką.

```
>QUE /START:QMG
```

Komenda uruchamia QMG i incjuje kolejkę domniemaną PRINT. Tworzy zbiór kolejek QUEUE.SYS w katalogu SY:[1,7], jeśli nie jest on tam obecny. Zeruje wszystkie wcześniej ustalone przypisania kolejek do urządzeń. Jeżeli zbiór

kolejek był właśnie aktywny, to wszystkie jego kolejki nadal zawierają swoje zlecenia, z tym, że kolejki chwilowo nie są przypisane do urządzeń, a ściślej - do zadań wyprowadzania systemowego związanych z urządzeniami /zlecenia wyprowadzania będą musiały być na nowo przyporządkowane do Podsystemu Wyprowadzania/.

6. Przyporządkować zadania wyprowadzania systemowego tj. zadania zainstalowane w kroku 2, do Podsystemu Wyprowadzania.

>QUE ddn: /SPOOL /FLAG:n /FORM:m

Na przykład:

>QUE LP: /SPOOL /FLAG:l /FORM:0

Ta komenda tworzy kolejkę LPQ0 i uruchamia poprzednio zainstalowane zadanie wyprowadzania LPPO. To zadanie przywiązuje sobie urządzenie LPO: 1 od tego momentu drukarka wchodzi pod kontrolę Podsystemu Wyprowadzania. Opisaną komendę należy wydać oddzielnie dla każdego z urządzeń wyprowadzania systemowego.

7. Przypisać kolejki do zadań wyprowadzania.

>QUE LP: /ASSIGN:PRINT

Komenda przypisuje kolejkę PRINT do zadania wyprowadzania LPPO. Zwykle robimy takie przypisanie tylko dla kolejki PRINT, bowiem kolejki związane (nazwa) z określonymi urządzeniami są do nich przypisane na mocy klucza /SPOOL, a więc już w kroku 6 procedury. Zatem przypisania dla innych kolejek nie są wymagane. Dla każdego z urządzeń wyprowadzania systemowego powołuje się kolejkę, ale kolejki te mogą być przypisywane również do innych urządzeń, jeżeliby zaistniała taka potrzeba. Np. jeżeli drukarka LPO: jest chwilowo niedostępna, to można przypisać kolejkę LPQ0 do drukarki LP1:.

Można przypisać kolejkę do więcej niż jednego zadania wyprowadzania i odwrotnie - po kilka kolejek do danego urządzenia i zadania wyprowadzania.

4.2. Przykład komend uruchomienia Podsystem Wyprowadzania

Niżej podano przykład zbioru komend SY:[1,2] QMGSTART.CMD.

```
INS DBO:QMG /PRI=70
INS DSO:LPP /PRI=70
INS DSO:QMGPRT
INS DSO:QMGCLI /TASK=...PRI
INS DSO:QMGCLI /TASK=...QUE
QUE /START:QMG
QUE LP: /SPOOL /FLAG:1 /FORM:0
QUE LP: /ASSIGN:PRINT
```

5. Opisy uprzywilejowanych komend QMG

Uruchomienie QMG odbywa się za pomocą pewnej liczby specjalnych funkcji realizowanych przez komendy uprzywilejowane.

Mają one następujący format:

```
>QUE ddnn: /function[ :option]...[/sw]...
ddnn:      Nazwa urządzenia, do którego odnosi się komenda.
/function  Określenie funkcji realizowanej przez komendę.
:option    W wierszu komendy może wystąpić jedna lub więcej opcji.
/sw        Określa jeden lub kilka kluczy dla funkcji lub opcji.
```

Opisywane komendy umożliwiają uprzywilejowanemu użytkownikowi: przyłączanie i odłączanie urządzeń Podsystemu Wyprowadzania, uruchamianie i zatrzymanie urządzeń, przypisywanie domniemanej kolejki do urządzeń, uruchamianie i zatrzymanie Programu Zarządzania Kolejką.

5.1. STOP

Funkcja STOP zawieszona urządzenie albo zatrzymuje Program Zarządzania Kolejką QMG.

Format

```
>QUE ddnn: /STOP[:option]
```


Opcje

:EOF
:EOJ
albo
>QUE /STOP:QMG

ddnn: Nazwa urzędzenia jest wymagana w przypadku opcji :EOF i :EOJ, natomiast nie podaje się jej dla opcji /STO:QMG, służącej do zatrzymania programu QMG.

Opcje Jeżeli wystąpi ddnn: i nie wystąpi żadna z opcji :EOF, :EOJ, wówczas określone urządzenie zostanie zawieszona natychmiast.

:EOF Urządzenie zatrzyma się po wyprowadzeniu bieżącego zbioru.

:EOJ Urządzenie zatrzyma się po zrealizowaniu bieżącego zlecenia drukowania.

:STO:QMG Powoduje zatrzymanie Programu Zarządzania Kolejką, po zrealizowaniu bieżącego zlecenia. Wszystkie przypisania kolejek do urządzeń zostaną unieważnione. Wszystkie zadania wyprowadzania zostaną odłączone od urządzeń.

Przykłady

>QUE LPO: /STOP

Urządzenie LPO: zostanie niezwłocznie zatrzymane. Niezależnie od tego do kolejki /kolejek/ związanej z tym urządzeniem mogą być wprowadzane nowe zlecenia.

QUE LP2: /STO:EOJ

Urządzenie LP2: zostanie zatrzymane po zrealizowaniu aktualnie aktywnego zlecenia.

QUE /STOP:QMG

Ta komenda zatrzymuje /kończy pracę/ QMG. QMG jest oznaczony do zatrzymania, ale bieżące zlecenie drukowania będzie zakończone.

5.2. START

Komenda START jest symetryczna do STOP. Uruchamia urządzenie uprzednio zawieszona komendą STOP, albo uruchamia QMG uprzednio zatrzymany komendą STOP.

Format

>QUE ddnn: /STArt[:option]...
[/sw]...

Opcje

Klucze

:NExt /FOrM:n

:TOP /FLag:n

:PAge:n

:BACk:n

:FORward:n

albo

>QUE /STArt:QMG

ddnn: Nie podaje się nazwy urządzenia w przypadku użycia opcji /STA:QMG.

Opcje

W przypadku komendy START pola opcji i kluczy wzajemnie się wykluczają. Stosuje się jedno bądź drugie. Jeżeli nie występują ani opcje, ani klucze, oznacza to pustą opcję. W tym przypadku urządzenie rozpoczyna drukowanie od miejsca, w którym zostało uprzednio zatrzymane.

:NExt Drukowanie rozpoczyna się od początku następnego zlecenia.

:TOP Drukowanie rozpoczyna się od początku bieżącego zbioru.

:PAge:n Drukowanie rozpoczyna się od n-tej strony bieżącego zbioru.

:BACk:n Drukowanie rozpoczyna się o n stron wcześniej niż strona, przy której nastąpiło zatrzymanie.

:FORward:n Drukowanie rozpoczyna się o n stron dalej niż strona, przy której nastąpiło zatrzymanie.

STA:QMG Te opcje stosuje się bez nazwy urządzenia ddnn:. Powoduje ona wykonanie następujących czynności:

- Uruchomienie Programu Zarządzania Kolejką QMG
 - Zainicjowanie domniemanej kolejki PRINT
 - Założenie zbioru SY:[1,7]QUEUE.SYS, jeżeli gą nie ma
 - Unieważnienie wszystkich przypisań poprzednio ustalonych
- W przypadku, gdy zbiór SY:[1,7]QUEUE.SYS był założony, wszystkie zapisane w nim kolejki muszą jednak na nowo otrzymać przypisania do zadań wypro-
wadzania.

Klucze

- W przypadku komendy /START pola opcji i kluczy wykluczają się wzajemnie. Stosuje się albo klucze, albo opcje.
- /Form:n Ten klucz zmienia typ formatu strony dla zadania wyprowadzania, w odniesieniu do typu ustalonego przez funkcje /SPOOL.
- /Flag:n Ten klucz zmienia ustawienie wskaźnika wyprowadzania stron winietowych, w odniesieniu do wskaźnika ustalonego przez funkcje /SPOOL. Zlecenia będą miały po n stron winietowych. Jeżeli użytkownik poda /FLA w kome-
dzie PRINT, wówczas każdy zbiór zlece-
nia będzie miał także n stron winieto-
wych. Wartość n może być 0, 1, 2 lub 3. Jeżeli używa się specjalnych formatów stron, to podaje się n=0, gdyż wówczas strony winietowe nie są wyprowadzane.

Przykłady

>QUE LPO: /STA
Drukarka LPO: znowi pracę od miejsca, w którym była zatrzymana.

>QUE LP2: /START:NE
Drukowanie rozpoczyna się od następnego zlece-
nia z kolejki /kolejek/ przypisanej do LP2:.

Uwagi

Jeżeli w komendzie PRINT definiującej zlecenie nie określono /Length:n to opcje komendy /START związane ze stronami są ignorowane.

Jeżeli zlecenie wyprowadzania jest restartowane w celu obsłużenia specjalnych formatów stron, to może pojawić się potrzeba ustawienia /FL na 0.

5.3. SPOOL

Funkcja ta powoduje przyłączenie urządzenia do Podsystemu Wyprowadzania. Zostaje nazwane i uruchomione zlecenie wyprowadzenia na to urządzenie, tworzy się także kolejka o nazwie związanej z nazwą urządzenia. Kolejka ta zostaje przypisana do zadania wyprowadzania na mocy działania tej komendy.

Format

>QUE ddn: /SPool [/sw]

Klucze

/FORM:n

/FLAG:n

/EXTERNAL

ddn: /SPool Powoduje utworzenie kolejki ddQn i uruchomienia zadania wyprowadzania ddPn, które powinno być pod tą nazwą wcześniej zainstalowane dla urządzenia ddn:. Przyłącza urządzenie ddn: do zadania wyprowadzania ddPn, udostępnionego dla zleceń z kolejki ddQn.

Klucze

/FORM:n

W przypadku tej funkcji można użyć jednocześnie kilku kluczy. Ustala, że zlecenie wyprowadzania na dane urządzenie ma realizować zlecenia drukowania z typem formatu o wartości n. Można spowodować przyjmowanie więcej niż jednego typu formatu, lecz wymaga to zdublowania defi-

nicji formatów. Szerszą informację na temat formatów wyprowadzania podane w rozdziale "Podsystem Wyprowadzania PRTQM" podręcznika "Programu konserwacji zbiorów".

/FLAG:n

Ustala ilość stron winietowych poprzedzających wydruk dla każdego zlecenia drukowania realizowane na danym urządzeniu. W przypadku używania specjalnych formatów może zajść potrzeba ustawienia n=0, aby uniknąć stron winietowych.

Klucz ten powoduje również, że dla każdego zbioru będzie wyprowadzone n stron winietowych zbioru, o ile w komendzie PRINT definiującej zlecenie podano klucz /FLAGS.

Wartością domniemaną n jest 0.

/External

Określa, że urządzenie, które ma być przyłączone do Podsystemu Wyprowadzania, nie jest częścią lokalnej konfiguracji. Klucz ten stosuje się wtedy, gdy należy wyprowadzić wyniki do zadania użytkowego, albo do sieci, a także w innych przypadkach używania Podsystemu do ładowania w dół systemu sieciowego.

Przykłady

>QUE LPO: /SPOOL

Ta komenda przyłącza LPO: do Podsystemu Wyprowadzania.

>QUE LP3: /SP /FO:2

Ta komenda przyłącza do Podsystemu Wyprowadzania drukarkę LP3:, z ustawieniem numeru formatu na 2.

5.4. USPOOL

Funkcja ta służy do usuwania urządzenia z listy urządzeń Podsystemu Wyprowadzania. Urządzenie

nie będzie już przypisane do kolejki. Zadanie wyprowadzania związane z tym urządzeniem zatrzymuje się, zaś kolejka o nazwie związanej z tym urządzeniem jest usuwana.

Format

>QUE ddnn: /UNSPool

Przykład

>QUE LPO: /UNSP

Ta komenda powoduje odłączenie urządzenia LPO: od zadania wyprowadzania LPPO i od Podsystemu Wyprowadzania.

5.5. ASSIGN

Funkcja ta tworzy połączenie pomiędzy kolejką a zadaniem wyprowadzania.

Format

>QUE ddnn: /ASsign:quename

Zwykle używa się ASSIGN do przypisania domniemanej kolejki PRINT do jednego lub więcej zadań wyprowadzania. Jest jednak możliwość przypisywania dowolnej kolejki do dowolnego zadania wyprowadzania.

Przykład

>QUE LPO: /AS:PRINT

Ta komenda przypisuje kolejkę PRINT do urządzenia LPO:.

5.6. DEASSIGN

Funkcja DEASSIGN unieważnia przypisanie kolejki do zadania wyprowadzania.

Format

>QUE ddnn: /DEAssign:quename

DEASSIGN eliminuje połączenie pomiędzy kolejką a zadaniem wyprowadzania, ale kolejka nadal istnieje, i wszystkie jej zlecenia nadal znajdują się w zbiorze kolejek. Co więcej, do kolejki mogą być dodawane nowe zlecenia. Będą one jednak realizowane dopiero po przypisaniu kolejki do urządzenia /do zadania wyprowadzania/.

6. Komunikaty Podsystemu Wyprowadzania

Komunikaty Podsystemu Wyprowadzania zaczynają się od znaków PRI lub QUE, zależnie od tego, z jakimi komendami użytkownika są związane.

Przykładowo:

PRI -- NO SUCH COMMAND

QUE -- DIRECTORY NOT ALLOWED

Zestawienie komunikatów znajduje się w rozdziale "Podsystem Wyprowadzania PRTQMG" podręcznika "Programy konserwacji zbiorów".

Oddzielnie należy wyjaśnić, że odpowiedź w postaci przynaglenia (>) na komendę QUE /LI (lub /BR lub /FU) oznacza, iż w systemie nie założono zbioru kolejek.

PROGRAM KOPIOWANIA I KOMPRESJI DYSKÓW

Program kopiowania i kompresji dysków (DSC) kopiuje zbiory przechowywane na dyskach o strukturze zbiorów systemu DOC PB na taśmie lub dysk; DSC odtwarza również zawartość tomów dyskowych, z utworzonych przez ten program taśmowych zbiorów archiwalnych. W czasie kopiowania DSC łączy zbiory i ich rozszerzenia w ciągłe obszary danych, co redukuje liczbę wskaźników bloków i nagłówków zbiorów na nowym tomie dyskowym.

Po skopiowaniu każdy zbiór na dysku wyjściowym zajmuje ciągły obszar, a miejsce dostępne dla nowych zbiorów jest również skonsolidowane w ciągły obszar na końcu nowego tomu dyskowego. DSC przeprowadza operacje kopiowania /oraz porównywania/ za pośrednictwem dwóch buforów. Każdy z tych buforów ma długość wystarczającą, aby zmieścić zawartość czterech bloków dyskowych oraz 16-to bajtowy nagłówek bufora. Używając klucza /BL można zwiększyć rozmiary buforów (o wielokrotność długości bloku), aż do maksymalnego rozmiaru, dopuszczalnego w danej wersji systemu.

1. Urządzenia obsługiwane przez DSC

Urządzenia obsługiwane przez program DSC, w wersji pracującej pod kontrolą systemu operacyjnego DOC PB, nie mogą być zamontowane. W tablicy 1 zebrane zostały urządzenia obsługiwane przez program DSC.

Tablica 1

Urządzenia obsługiwane przez DSC

Mnemonik	Nazwa dysku	Typ
DB:	SMRP4/5/6 - dysk pakietowy	Struktura blokowa
DF: (*)	SMRF/SMRS - dysk z nieru- chomymi głowicami	Struktura blokowa

DK:	SMRK5/5F - dysk kasetowy	Struktura blokowa
DL:	SMRL1/2 - dysk kasetowy wymienny	Struktura blokowa
DM:	SMRK6/7 - dysk kasetowy wymienny	Struktura blokowa
DP:	SMRP2/3 - dysk pakietowy	Struktura blokowa
DR:	SMRM2/3/SMRP7 - dysk pakietowy	Struktura blokowa
DS: (*)	SMRS3/4 - dysk i nieruchomymi głowicami	Struktura blokowa
DX: (*)	SMRX - dysk elastyczny	Struktura blokowa
DY: (*)	SMRX2 - dysk elastyczny	Struktura blokowa
MM:	SMTM2/3-SMTE16/SMTU16/45/77	Taśma 9-ścieżkowa taśma magnetyczna
MT:	SMTU/10/SMTE10	Taśma magnetyczna

2. Wywołanie i zakończenie DSC w wersji pod systemem

Program DSC może być wywoływany, tak jak inne zadania użytkowe w systemie DOC PB. Aby zakończyć działanie programu DSC należy wprowadzić znak końca zbioru <CTRL/Z>.

2. Wywołanie i zakończenie samodzielnej wersji DSC

Program DSC w wersji samodzielnej (DSCS8.SYS) można wywołać z dysku lub z taśmy, dostarcza nej z systemem operacyjnym.

W systemie DOC PB można wywołać wersje samodzielną DSC dwoma równoprawnymi drogami:

1. Używając uprzywilejowanej komendy BOO:
BOO [1,54]DSCS8.SYS

(*) Urządzenie nie jest obsługiwane w samodzielnej wersji programu DSC

Komenda ta powoduje załadowanie do pamięci operacyjnej samodzielnej wersji DSCS8.SYS, ze zbioru na dysku LB:, z katalogu [1,54]

2. Montując taśmę magnetyczną zawierającą obraz DSCS8.SYS na jednostce pamięci taśmowej, a następnie ładując program do pamięci operacyjnej (używając kluczy maszyny).

Wersja samodzielna DSC po załadowaniu i wystartowaniu wypisuje na terminalu TTO: następujący komunikat:

```
DOC PB VO2 BL26 DISK SAVE AND COMPRESS UTILITY  
/DOC PB VO2 BL26 program kopiowania i kompresji  
dysków/ DSCS8>
```

i jest gotowa do przyjęcia komendy. Działanie DSC w wersji samodzielnej można zakończyć za-trzymując procesor kluczem.

4. Format komendy programu DSC

Komendy programu DSC mają następujący format:
DSC>outdev[s]:[fillab][/sw]=indev[s]:[fillab]
[/sw]

gdzie:

Parametry wyjściowe:

outdev: oznacza urządzenie /urządzenia/ fizyczne, do którego kopiowane są dane. Jeżeli wśród parametrów wyjściowych, jest więcej niż jedna jednostka taśmy magnetycznej, to symbole urządzeń muszą występować po kolei, oddzielone przecinkami. Jeżeli DSC zapisuje zawartość dysku na kilku tomach taśmy magnetycznej, to używa podane jednostki kolejno, w trybie okrężnym.

fillab jest identyfikatorem taśmowego zbioru archiwalnego tworzonych przez DSC na taśmie magnetycznej lub nazwą wyjściowego tomu dyskowego. Jeżeli etykieta ta nie jest specyfikowana, to jako identyfikator przyjmowana jest nazwa wejściowego tomu dyskowego.

/sw oznacza jeden lub więcej opcjonalnych kluczy

Parametry wejściowe:

indev: oznacza wejściowe urządzenie /urządzenia/ fizyczne w formacie AA[nn]:, z którego dane kopiowane są na urządzenie wyjściowe

fillab jest identyfikatorem utworzonego przez DSC zbioru taśmowego, który jest kopiowany na dysk lub jest z nim porównywany. Jeżeli etykieta nie jest wyspecyfikowana, gdy dysk jest odtwarzany z taśmy magnetycznej, to kopiowany jest pierwszy zbiór archiwalny z wejściowej taśmy magnetycznej.

Uwaga

Każdy zbiór zapisany w formacie taśmowym DSC, jest archiwalną kopią całej zawartości tomu dyskowego o strukturze zbiorów systemu DOC PB

/sw oznacza jeden lub więcej opcjonalnych kluczy

5. Deklaracja etykiet zbiorów i klucze programu DSC

W komendzie programu DSC mogą wystąpić deklaracje zbiorów oraz klucze. Część kluczy ma opcje, które również mogą być używane w komendzie. W tabelicy 2 zawarty jest krótki opis kluczy i ich funkcji. Wszystkie opisane tutaj klucze są dostępne w obu wersjach programu DSC. Klucze dostępne jedynie w samodzielnej wersji programu DSC, opisane są w tabelicy 3.

Tablica 2

Opis kluczy programu DSC

Nazwa klucza	Format klucza	Opis klucza
klucz dopisywania	/AP	dopisuje zbiór archiwalny do tomu taśmowego. Dopisywany zbiór staje się ostatnim na taśmie wyjściowej
klucz bloków uszkodzonych	MAN /BAD= NOAUTO MAN:NOAUTO	pozwała na manualne wskazanie bloków uszkodzonych. Możliwe jest: uzupełnienie, zmiana lub zignorowanie zbioru opisów bloków uszkodzonych określa ilość bloków o długości 256 słów, składających się na wewnętrzne buforu programu DSC
klucz współczynnika blokowania	/BL	inicjuje porównanie zawartości urządzenia wejściowego i wyjściowego
klucz porównania	/CMP	zmienia standardową gęstość zapisu na taśmach magnetycznych (wynoszącą 800 BPI). Opcja /DENS=1600 powoduje reakcje tomu taśmowego zapisanego z gęstością 1600 BPI. Opcja /DENS=800:1600 /gęstość zmienna/ powoduje reakcje
klucz gęstości	/DENS=1600 /DENS=800:1600	

nowego tomu taśmowego z nagłówkiem tomu zapisanym z gęstością 800 BPI, natomiast pozostałe zapisy na taśmie wykonane są z gęstością 1600 BPI.

Należy zaznaczyć, że klucz gęstości w wersji /DENS=1600 można używać dla pamięci taśmowych SMTU16,

SMT16 oraz SMTU45.

Klucz gęstości w wersji /DENS=800:1600 można używać dla pamięci taśmowych SMTU16 oraz SMTU45, jeżeli nie współpracują one z formatem SMTE3

klucz /RW
przewinięcia

powoduje przewinięcie wszystkich taśm przed wykonaniem komendy /kopiowania lub porównania/

klucz /VE
sprawdzania

inicjuje skopiowanie tomu dyskowego lub zbioru archiwalnego ze sprawdzeniem poprawności kopiowania

5.1. Klucz sprawdzenia (/VE)

Klucz sprawdzenia /VE podawany w specyfikacji wyjściowej, inicjuje kopiowanie tomu a po skopiowaniu sprawdzenie, czy kopiowanie było poprawne. DSC nie dopuszcza deklaracji kluczy /VE lub /CMP wtedy, gdy urządzeniem wejście - wym i wyjściowym jest pamięć taśmowa.

Jeżeli pamięć taśmowa jest urządzeniem wejściowym lub wyjściowym, to operacja sprawdzenia wykonywana jest po wypełnieniu każdego tomu taśmowego zbioru archiwalnego. Na przykład: DSC wypełnia taśmę MM1: i sprawdza tę taśmę, następnie wypełnia MM2: i również ją sprawdza, powtarzając ten cykl aż do przepisania zawartości dysku. W czasie kopiowania dysku na inny dysk sprawdzanie rozpoczyna się po zakończeniu kopiowania.

Format komendy zawierającej klucz /VE jest następujący:

```
outdev:[fillab]/VE=indev:[fillab]/sw
```

Jeśli w komendzie nie podano etykiety zbioru dla wejściowego taśmowego zbioru archiwalnego, to klucz /VE inicjuje kopiowanie od pierwszego zbioru na taśmie.

Jeśli DSC wykrywa w czasie kopiowania EOF lub EOV na jednym z tomów taśmowego zbioru archiwalnego, to przesuwa taśmę na początek bieżącego zbioru /lub na początek tomu/ i rozpoczyna sprawdzanie kopiowania.

Jeśli w czasie weryfikacji DSC wykrywa EOV dowolnym tomie lub EOF na tomie, który nie jest pierwszym tomem zbioru archiwalnego, to przewija taśmę na początek i rozładowuje jednostkę taśmową. Jeśli wykryto znacznik EOF, to po rozładowaniu jednostki DSC rozpoczyna kopiowanie następnego tomu zbioru /od początku nośnika/.

6. Klucze wersji samodzielnej programu DSC

Wersja samodzielna programu DSC (DSCS8.SYS) nie ma wszystkich własności wersji standardowej. W wersji samodzielnej DSC nie są obsługiwane dyski elastyczne oraz dyski z nieruchomymi głowicami DF: i DS:.

Jeśli program DSC w wersji samodzielnej wykonany będzie bez użycia kluczy opisywanych w dalszej części niniejszego punktu, to przyjęty zostanie następujący tryb pracy programu DSC:

- logiczne numery jednostek są ograniczone do liczb 0 i 1
- nie więcej niż dwa urządzenia wyjściowe lub wejściowe mogą być wyspecyfikowane w jednej komendzie
- wyłącznie jeden kontroler może być używany dla każdego typu urządzenia
- adresy wektorów przerwań dysków pakietowych SMRP2 i SMRP3, dysków z nieruchomymi głowicami SMRS3 i SMRS4 oraz jednostek taśmy magnetycznej SMTU10 i SMTS3, mają niestandardowe wartości

Można w pewnym stopniu, ominąć ograniczenia nakładane przez użycie wersji samodzielnej DSC, poprzez użycie kluczy przedstawionych w tablicy 3, przy pomocy których można zmieniać wartości standardowe na odpowiadające aktualnej konfiguracji sprzętu. Klucze te mogą być używane wyłącznie w wersji samodzielnej FSC.

Tablica 3

Klucze wersji samodzielnej programu DSC

Format klucza	Nazwa klucza	Opis
/CSR=xxxx	klucz adresu kontrolera (CSR)	klucz zmienia adres rejestru CSR kontrolera podanego typu urządzenia
/TMO2=x	klucz formatera	klucz zmienia numer formatera
/UNIT=x	klucz jednostki	klucz zmienia numer jednostki fizycznej urządzenia
/VEC=xxx	klucz adresu przerwania	klucz zmienia adres wektora przerwania

Powyższe cztery klucze dla wersji samodzielnej DSC mogą występować łącznie w pojedynczej komendzie, w celu zmiany potrzebnych wartości dla pojedynczego urządzenia lub typu urządzenia.

PROGRAM ARCHIWACJI TOMÓW

Program archiwacji tomów (BRU) umożliwia przechowywanie i odtwarzanie zawartości dysków zapisanych zgodnie ze standardem podsystemu zbiorów systemu DOC PB. Program kopiuje zawartość tomu dyskowego do tomu archiwalnego /lub tomów/ umożliwiając przechowanie informacji i odtworzenie ich w przypadku zniszczenia (usunięcia zbioru, uszkodzeniu nośnika, itp.).

Archiwacja i odtwarzanie mogą być wykonywane między dyskami i taśmami magnetycznymi. Dopuszczalne są następujące operacje:

- archiwacja informacji z dysku na taśmie magnetycznej
- odtwarzanie informacji z taśmy magnetycznej na dysku
- kopiowanie informacji między dyskami, dla ich przechowania i odtwarzania

Oprócz powyższych operacji bazowych, dostępne są operacje sterujące, reprezentowane przez następujące komendy:

- komendy archiwacji lub odtwarzania informacji, wraz ze specyfikacją zbiorów, dat i czasu ich utworzenia
- komenda inicjacji dysku przed operacją kopiowania lub odtwarzania informacji. Możliwe jest nadawanie dyskowi nazwy /podobnie jak czyni to program INI/ oraz tworzenie opisu uszkodzonych bloków /podobnie jak czyni to program BAD/
- komendy sterujące archiwowaniem na taśmie magnetycznej. Możliwe jest etykietowanie taśmy, ustalanie gęstości zapisu oraz operacje przewijania i dopisywania /dopisywanie informacji na końcu taśmy/
- komendy weryfikacji poprawności zapisu informacji
- komendy drukowania informacji o zawartości nośników archiwalnych

1. Klucze komend i ich domniemane wartości

W tablicy 1 przedstawiony został ogólny opis kluczy komend, realizujących operacje archiwacji i odtwarzania zbiorów.

Tablica 1

Zestawienie komend programu BRU

Klucz komendy	Opcje	Domniemana wartość
/APPEND		nie ma
/BACKUP_SET:nam		nie ma
/BAD:	[MANUAL] [AUTOMATIC] [OVERRIDE]	BAD:AUTOMATIC
/BUFFERS:nmb		ilość buforów na dysku oryginalnym
/COMPARE		
/CREATED:	[BEFORE:(dd-mmm-yy [hh:mm:ss])] [AFTER:(dd-mmm-yy [hh:mm:ss])]	bieżąca data
/DENSITY:nmb		DENSITY:800.
/DIRECTORY		nie ma
/DISPLAY		nie ma
/ERRORS:nmb		ERRORS:25
/EXCLUDE		nie ma
/EXTEND:nmb		ilość bloków na dysku oryginalnym
/HEADERS:nmb		ilość nagłówek na dysku oryginalnym
/INITIALIZE		nie ma
/INVOLUME:nam		nie ma
/LENGTH:nmb		długość taśmy wyjściowej
/MAXIMUM:nmb		maksymalna ilość zbiorów na dysku oryginalnym
/MOUNTED		nie ma
/NEW_VERSION		nie ma

/NOINITIALIZE		nie ma
/NOPRESERVE		nie ma
/NOSUPERSEDE		nie ma
/OUTVOLUME:nam		nazwa dysku oryginalnego
/POSITION:	[BEGINNING]	zbiór nagłówków
	[MIDDLE]	umieszczony tak
	[END]	jak na dysku
	[BLOCK:nmb]	oryginalnym
/PROTECTION:	SYSTEM:[R][W][E][D]	
	OWNER:[R][W][E][D]	
	GROUP:[R][W][E][D]	
	WORLD:[R][W][E][D]	
/REVISED:	[BEFORE:(dd-mmm-yy [hh:mm:ss])]	bieżąca data
	[AFTER:(dd-mmm-yy [hh:mm:ss])]	
/REWIND		nie ma
/SUPERSEDE		/NOSUPERSEDE
/TAPE_LABEL:lbl		nie ma
/VERIFY		nie ma
/WINDOWS:nmb		liczba wskaźni- ków mapowania taka, jak na dysku oryginal- nym

1.1. Komendy selektywnej archiwacji i odtwarzania

Komendy opisane w tablicy 2 pozwalają archiwować o odtwarzać zbiory danych. Możliwe jest selektywne przemieszczanie zbiorów między nośnikiem archiwalnym o oryginałem, przez podanie następujących kryteriów wyboru zbiorów do przepisywania:

- specyfikacji zbioru
- daty i czasu utworzenia zbioru
- daty i czasu ostatniego poprawiania zbioru

Tablica 2

Komendy selektywnej archiwacji i odtwarzania

Klucz komendy	Opcje	Opis
/CREATED:	[BEFORE:(dd-mmm-yy [hh:mm:ss])] zbiorów utworzonych przed [AFTER:(dd-mmm-yy [hh:mm:ss])] (opcja BEFORE) lub po (opcja AFTER) podanej w komendzie dacie i/lub godzinie	
/EXCLUDE		klucz pominięcia spośród przenoszonych zbiorów wszystkich, które wymieniono w wierszu komendy
/NEW_VERSION		klucz powoduje, że w wypadku identycznej specyfikacji zbiorów, w tomie wyjściowym tworzony jest zbiór o większym numerze wersji i obie wersje będą zachowane
/NOSUPERSEDE		klucz zapewnia, że w tomie wyjściowym nie będą usuwane istniejące wersje zbioru, w

/REVISED: [BEFORE:(dd-mmm-yy
[hh:mm:ss])]
[AFTER:(dd-mmm-yy
[hh:mm:ss])]

/SUPERSEDE

przypadku kopiowania do tomu wyjściowego, zbioru o specyfikacji istniejącej już w tym tomie. Zbiór z nośnika wejściowego nie jest kopiowany
przeniesienie zbiorów poprawianych przed (opcja BUFORE) lub po (opcja AFTER) podanej w kolumnie dacie i/lub godzinie
klucz zapewnia, że w tomie wyjściowym, będą usuwane istniejące wersje zbioru, w przypadku kopiowania do tomu wyjściowego, zbioru o specyfikacji istniejącej już w tomie. Konflikt ten będzie rozwiązany

przez usunięcie zbioru istniejącego już w tomie wyjściowym i zapisanie zbioru z nośnika wejściowego

Niżej wymienione komendy określają sposoby rozwiązywania konfliktów powstających w czasie operacji odtwarzania zbiorów, w przypadku wystąpienia na tomie wejściowym specyfikacji zbioru identycznej, ze specyfikacją jednego ze zbiorów w tomie wyjściowym:

- /NEWVERSION
- /SUPERSEDE
- /NOSUPERSEDE

1.2. Komendy sterujące obsługą dysków

Komendy przedstawione w tablicy 3 umożliwiają sterowanie sposobem, w jaki system operacyjny obsługuje dysk w czasie przenoszenia zbiorów.

Tablica 3

Komendy sterowania obsługą dysków

Klucz komendy	Opcje	Opis
/INITIALIZE		klucz inicjacji dysku wyjściowego przed rozpoczęciem przepisywania zbiorów. Klucz ten dotyczy odtwarzania zbiorów dyskowych
/INVOLUME:	nam	specyfikacja nazwy tomu dysku wejściowego
/NOINITIALIZE		klucz wskazuje, że dysk wyjściowy nie będzie inicjowany oraz że zawiera on już strukturę zbiorów systemu operacyjnego DOC PB i jest zamontowany

<code>/MOUNTED</code>		klucz wskazuje, że dysk wejściowy zawiera zbiory w standardzie systemu DOC PB i jest zamontowany
<code>/NOPRESERVE</code>		klucz wskazuje, że BRU nie musi zachować identyfikatorów zbiorów
<code>/OUTVOLUME :</code>	nam	specyfikacja nazwy tomu dysku wyjściowego

1.3. Klucze sterujące obsługą taśmy

Zawarte w tablicy 4 klucze komend, umożliwiają sterowanie zapisywaniem /i odtwarzaniem / zbiorów na taśmach magnetycznych.

Tablica 4

Komendy sterowania obsługą pamięci taśmowych

Klucz komendy	Opcje	Opis
<code>/APPEND</code>		klucz dopisania nowego zbioru archiwalnego, na taśmie zawierającej jeden lub kilka taśmowych zbiorów archiwalnych
<code>/BACKUP_SET:</code>	nam	klucz określa nazwę zbioru archiwalnego, który ma być zapisany na taśmie
<code>/DENSITY:</code>	nmb	klucz określa gęstość, z jaką BRU ma zapisać zbiór na taśmie
<code>/ERRORS:</code>	nmb	klucz określa dopuszczalną ilość usuwalnych błędów we/wy, podczas odczytu z pamięci taśmowej. Przekroczenie tej liczby powoduje przerwanie operacji
<code>/LENGTH:</code>	nmb	klucz określa długość wyjściowej taśmy magnetycznej /dziesiętnie w stopach/

- /REWIND klucz powoduje przewinięcie na początek pierwszego tomu wielotomowego, taśmowego zbioru archiwalnego
- /TAPE_LABEL: 1b1 klucz określa 6-cio znakową etykietę taśmy /według standardu ANSI/

1.4. Klucze weryfikacji

Klucze zawarte w tablicy 5 umożliwiają wykrywanie różnic między informacjami zawartymi na urządzeniu wejściowym i wyjściowym.

Tablica 5

Komendy weryfikacji

Klucz komendy	Opis
/COMPARE	klucz powoduje porównanie danych na urządzeniu wyjściowym z danymi na urządzeniu wejściowym i wydrukowanie raportu o wykrytych różnicach
/VERIFY	klucz inicjuje kopiowanie danych z urządzenia wejściowego na wyjściowe a następnie porównanie skopiowanych danych i wydrukowanie raportu o wykrytych różnicach

1.5. Klucze listowania

Klucze zawarte w tablicy 6 powodują drukowanie informacji o zbiorach, które zostały archiwizowane.

Tablica 6

Komendy drukowania

Klucz komendy	Opis
/DISPLAY	klucz powoduje drukowanie na terminalu katalogu oraz specyfikacji każdego zbioru, który kopiowany jest przez program BRU

/DIRECTORY klucz powoduje drukowanie na terminalu informacji o zawartości taśmy magnetycznej. Drukowana jest nazwa taśmowego zbioru archiwalnego, nazwa zbioru danych oraz numer tomu taśmowego zbioru archiwalnego

1.6. Klucze inicjacji dysku

Podając klucz /INI oraz któryś z kluczy zawartych w tablicy 7, można sterować inicjacją dysku wyjściowego.

Tablica 7

Komendy inicjacji dysku

Klucz komendy	Opcje	Opis
/BAD:	MANUAL AUTOMATIC OVERRIDE	określa sposób wprowadzania opisu bloków uszkodzonych na dysku
/BUFFERS:	nmb	określa ilość katalogów /dla danego tomu/, których bloki zarządzania zbiorem (FCB), zadanie ACP ma przechować w pamięci
/EXTEND:	nmb	określa moduł rozszerzania zbioru /w blokach/, o które zbiory danego tomu mają być rozszerzane
/HEADERS:	nmb	określa liczbę nagłówków zbiorów, które wstępnie będą przydzielone w zbiorze nagłówków
/MAXIMUM:	nmb	określa maksymalną liczbę zbiorów, które można umieścić w tomie dyskowym

- /POSITION: BEGINNIG określa na dysku pozycje zbioru nagłówków katalogu głównego (MFD) oraz zbiory mapy zajętości
- /PROTECTION: SYSTEM:[R][W][E][D] określa glo -
OWNER:[R][W][E][D] balne kody
GROUP:[R][W][E][D] ochrony zbioru
WORLD:[R][W][E][D] rów na dysku
- /WINDOWS: nmb określa liczbę wskaźników mapowania używanych w oknach zbioru

1.7. Urządzenia obsługiwane przez program BRU

W tablicy 8 przedstawiono urządzenia obsługiwane przez program BRU.

Tablica 8

Urządzenia obsługiwane przez program BRU

Mnemo- nik	Nazwa dysku	Typ
DB:	SMRP4/5/F - dysk pakietowy	Struktura blokowa
DF:	SMRF/SMRS - dysk z nieruchomymi głowicami	Struktura blokowa
DK:	SMRK5/5F - dysk kasetowy	Struktura blokowa
DL:	SMRL/SMRL1 - dysk kasetowy wymienny	Struktura blokowa
DM:	SMRK6/7 - dysk kasetowy wymienny	Struktura blokowa
DP:	SMRP2/3 - dysk pakietowy	Struktura blokowa
DR:	SMRM2/3 - dysk pakietowy	Struktura blokowa
DS:	SMRS3/4 - dysk z nieruchomymi głowicami	Struktura blokowa
DX:	SMRX - dysk elastyczny	Struktura blokowa
DY:	SMRX2 - dysk elastyczny	Struktura blokowa

MM:	SMTM2/3-SMTE16/SMTU16/45/77	Taśma
	9-ścieżkowa taśma magnetyczna	
MT:	SMTU10/SMTE10 lub SMTS3 -	Taśma
	- 9-ścieżkowa taśma magnetyczna	
MS:	SMTS4 - taśma magnetyczna	Taśma

2. Format wiersza komendy programu BRU

W punkcie niniejszym opisany zostanie format wiersza komendy programu BRU. Opisane będą reguły syntaktyczne, parametry wiersza komendy, klucze komend oraz przynaglenia.

2.1. Opis wiersza komendy

Wiersz komendy programu BRU ma następujący format:

```
BRU /sw indev,...[filspc,...] outdev,...  
    [filspc,...]
```

Uwaga

Należy podkreślić odmiennność formatu komend programu BRU w stosunku do innych systemowych programów pomocniczych. W wierszu komendy najpierw należy podawać klucze komendy, dalej specyfikacje zbiorów wejściowych, a po nich oddzielone spacją (<SP>) specyfikacje zbiorów wyjściowych

2.2. Opis przynagleń

W programie BRU drukowane są następujące przynaglenia:

```
FROM:  
/skąd/
```

```
TO:  
/dokąd/
```

```
INITIALIZE [Y/N]:  
/inicjacja [Y/N]/
```

```
FROM polecenie podania nazwy /lub nazw/ urzą-  
dzeń, na których ma być zapisany tom wyj-  
ściowy lub zbiór archiwalny.
```

TO polecenie podania nazwy /lub nazw/ urządzeń, na których ma być zapisany to wyjściowy lub zbiór archiwalny.

INITIALIZE [Y/N] pytanie o to, czy użytkownik chce inicjować tom wyjściowy, niszcząc informacje dotychczas znajdujące się na nim. Jeśli tak, to należy podać odpowiedź (Y), co znaczy TAK (YES). Jeśli użytkownik chce jedynie dopisać zbiory w tomie wyjściowym należy podać odpowiedź (N), co znaczy NIE (NOT)

2.3. Opis parametrów wierszy komend

/sw specyfikacja kluczy komend, które przedstawiono w poprzedniej części opisu. Jeśli w wierszu komendy podano więcej niż jeden klucz, to muszą one być podane bez znaków rozdzielających, zatem nie mogą być rozdzielone znakami spacji (<SP>) ani żadnym innym znakiem. Klucze mogą występować w dowolnej kolejności.

Nazwa klucza może być dowolnie skracana, tak aby skrócona nazwa była unikalna w stosunku do nazw innych kluczy /oraz skrótów tych nazw/. Na przykład:

BRU)/REW/INI/OUT:BACKUP MMO: DKO:

BRU - STARTING TAPE 1 ON MMO:

BRU -- *WARNING* -- THIS DISK WILL NOT A HARDWARE BOOTABLE SYSTEM

BRU - END OF TAPE 1 ON MMO:

BRU - COMPLETED

Jeśli klucz ma opcjonalne parametry, to należy oddzielić od nazwy klucza znakiem (:), jak to niżej przedstawiono:

/sw:opcja

inpdev specyfikacja urządzenia wejściowego /lub urządzeń wejściowych/, z którego dane są kopiowane. Jeśli urządzeniem wejściowym jest pamięć taśmowa, to w wierszu komendy można podać kilka jednostek taśmowych. Specyfikacja urządzenia ma format:

dd[nn]:

gdzie dd jest dwuznakową nazwą urządzenia, a nn oktalnym numerem jednostki. Numer jednostki jest jedno lub dwucyfrową liczbą oktalną. Domniemanym numerem jednostki jest 0. Znak (:) jest znakiem ograniczającym nazwę urządzenia, a zatem podanie go w wierszu komendy jest obligatoryjne. Jeśli w wierszu komendy występuje wiele urządzeń, to ich mnemoniki muszą być rozdzielone przecinkami (,)

filspc specyfikacja zbioru, dla wskazania, który zbiór lub grupa zbiorów, ma być archiwowana lub odtworzona. Specyfikacja zbioru ma następujący format:

[n,m]filnam.typ;ver

W wierszu komendy można podać do 16-tu specyfikacji zbiorów.

Selektywne archiwowanie lub odtwarzanie można przeprowadzić wskazując katalog (UFD), nazwę zbioru, typ lub numer wersji. Jeśli kopiowanie selektywne względnie wersje zbioru, to numer wersji powinien być podany w jawnej postaci. Nie podanie numeru wersji lub zastosowanie konwencji gwiazdka (*), powoduje kopiowanie wszystkich wersji zbioru. Program BRU nie dopuszcza numerów wersji 0 oraz -1.

outdev specyfikacja urządzenia wyjściowego, na które dane będą zapisywane. Jeśli urzą-

dzeniem wyjściowym jest pamięć taśmowa, to w wierszu komendy można podać kilka jednostek taśmowych. Format specyfikacji urządzeń wyjściowych jest taki sam, jak opisany wyżej format specyfikacji urządzeń wejściowych.

2.3.1. Komenda bez specyfikacji zbiorów

Podanie wiersza komendy bez specyfikacji zbiorów, powoduje kopiowanie wszystkich zbiorów znajdujących się w wejściowym tomie dyskowym lub wejściowym zbiorze archiwalnym.

2.3.2. Konwencja gwiazdka

Konwencja gwiazdka może być stosowana tak, jak to opisano niżej:

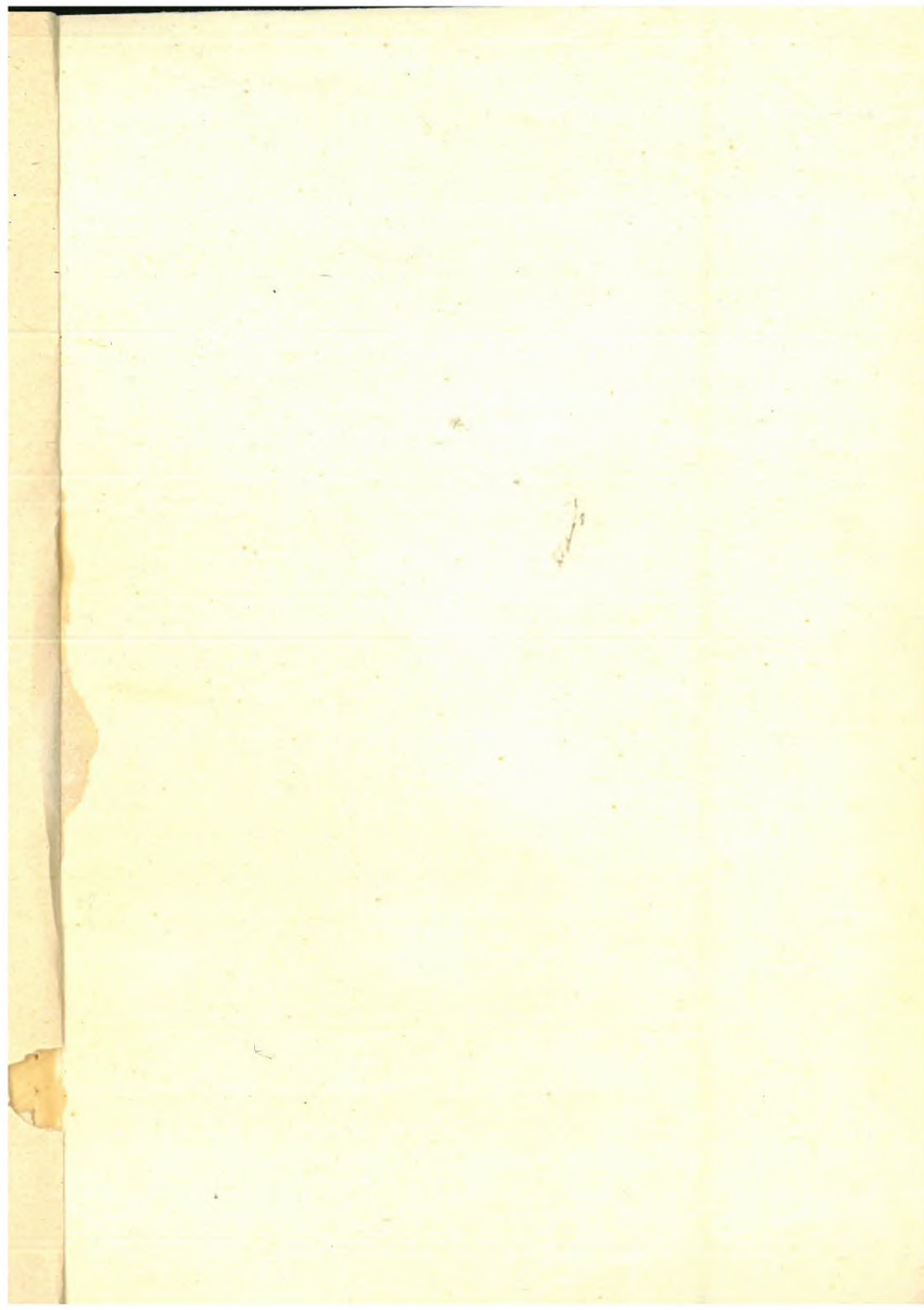
- [*,*] oznacza wszystkie kombinacje grup oraz użytkowników
- [n1,*] oznacza wszystkich użytkowników z grupy n1
- [*,n2] oznacza wszystkich użytkowników o numerze n2, z wszystkich grup

Program BRU realizuje konwencję gwiazdka (*) dla wszystkich elementów specyfikacji zbioru: nazwy zbioru, typu oraz numeru wersji. Realizowane są wszystkie reguły tej konwencji, zgodnie z zasadami przyjętymi w systemie operacyjnym DOC PB, poza następującymi wyjątkami:

- jeśli pominięto element specyfikacji, to program BRU stosuje do pominiętego elementu specyfikacji konwencję gwiazdka.
- jeśli w specyfikacji zbioru podano katalog, to kopiowane będą /w czasie archiwacji lub odtwarzania/ wszystkie zbiory należące do podanego katalogu /podanych katalogów/.

Universitas zam.114/85 n.230

H - 71



3337
9.90