

HAMID AL RAIMI

## REZYDENTNY PROGRAM KORZYSTANIA Z BAZY DANYCH W SYSTEMIE OPERACJNYM DOS

**Streszczenie.** Podczas pracy na komputerze często powstaje potrzeba skorzystania z różnych informacji (usług) niedostępnych w aktualnie realizowanym programie. Jednym z rozwiązań możliwych w systemie DOS jest zastosowanie programu rezydentnego. W artykule przedstawiono przykładowe rozwiązanie programu rezydentnego pozwalającego na korzystanie z zasobów bazy danych utworzonej na dysku. Omówiono sposób jego budowy oraz niektóre ogólne problemy, jakie można napotkać podczas tworzenia programów rezydentnych.

## A RESIDENT PROGRAM SERVICING DATABASE IN DOS

**Summary.** During work with a computer it is often necessary to use different information inaccessible in the executed program. One of possible solutions in DOS is application of a resident program. The exemplary resident program allowing to use database resources stored in the disk has been presented in the paper. A procedure of creating its structure and some general problems met during realization of such programs have been discussed as well.

### 1. OGÓLNE PROBLEMY BUDOWY I WYKORZYSTANIA PROGRAMÓW REZYDENTNYCH W SYSTEMIE DOS

Programem rezydentnym nazywamy tego rodzaju program, który po uruchomieniu pozostaje w pamięci operacyjnej również po przekazaniu sterowania do programu wywołującego (zwykle jest to system operacyjny).

Od tego momentu program rezydentny może w dowolnej chwili zostać uaktywniony w celu udostępnienia świadczonych przez niego usług. Zatem zadania realizowane przez program rezydentny są dostępne także podczas używania dowolnego innego programu. Nie zachodzi więc konieczność zakończenia pracy aktualnie wykonywanego programu w przypadku

chwilowej potrzeby skorzystania z programu rezydentnego. Tego typu rozwiązania są sensowne oczywiście jedynie w przypadku pracy z systemem operacyjnym nie udostępniającym wieloprocesowości. Takim właśnie systemem jednozadaniowym jest system DOS. Wszystkie dalsze rozważania dotyczą tego właśnie systemu.

Przy tworzeniu programów rezydentnych powstają specyficzne problemy nie występujące w programach innych typów. Dotyczą one przede wszystkim takich zagadnień, jak sposób uaktywnienia programu i wykrywania jego obecności w pamięci, ograniczeń co do momentu wywołania programu oraz zasady korzystania z takich struktur systemowych jak stos i blok wstępny programu [1].

### **Sposób uaktywnienia programu**

Można wskazać dwa podstawowe sposoby wywołania programu rezydentnego. Może być on uaktywniany za pomocą określonego klawisza (ang. hotkey) lub co pewien okres czasu. W obydwu przypadkach należy „podpiąć się” do procedur obsługi odpowiednich przerwań. W pierwszym przypadku (uaktywnienie przez klawisz) należy zmienić procedurę obsługi przerwań związanych z klawiaturą. W systemie DOS występują dwa tego rodzaju przerwania. Pierwsze z nich o numerze 16 dotyczy obsługi klawiatury na poziomie BIOS-a. Wykorzystanie tego przerwania do uaktywniania programu rezydentnego może jednak spowodować, że nie zawsze będzie to możliwe. Wynika to z faktu, że część programów omija procedury przerwań BIOS-a przy korzystaniu bezpośrednio z niższych warstw systemu. Także w przypadku gdy program przez dłuższy czas nie spodziewa się danych z klawiatury i nie wywołuje przerwań numer 16, naciśnięcie klawisza nie spowoduje natychmiastowego uaktywnienia programu rezydentnego, lecz zostanie ono „odłożone” do chwili wywołania tego przerwania przez aktualnie pracujący program. Problemy te nie występują, jeżeli do uaktywniania programu rezydentnego zostanie użyte przerwanie numer 9. Przerwanie to jest generowane po każdorazowym naciśnięciu lub zwolnieniu klawisza. Jednak aby zastosować tego rodzaju inicjację programu rezydentnego, należy napisać nową procedurę obsługi tego przerwania.

Drugim sposobem aktywizacji programu rezydentnego jest wykorzystanie przerwania związanego z zegarem systemowym (przerwanie 1CH). Przerwanie to jest wywołane przez procedurę BIOS-a obsługi zegara (przerwanie 8H) około 18,2 raza na sekundę. Można zatem napisać odpowiednią procedurę i jej adres wpisać do tablicy przerwań. Procedura taka będzie okresowo wywoływana przez BIOS. W ramach tej procedury można sprawdzać, czy upłynął określony odstęp czasu i jeśli tak jest, następuje inicjacja programu rezydentnego.

### **Sposób wykrycia obecności programu w pamięci**

Po wyjściu z programu rezydentnego znajduje się on w stanie nieaktywnym w pamięci komputera oczekując na inicjację ze strony użytkownika. Ponowne uruchomienie programu

w sposób standardowy [2] tworzy dodatkową kopię w pamięci niepotrzebnie ograniczając jej pojemność. Aby tego uniknąć, należy sprawdzić, czy program rezydentny znajduje się już w pamięci.

Pierwszą metodą sprawdzenia jest określenie, którym programom przypisane są kolejne bloki pamięci. Druga metoda opiera się na zastosowaniu przerwania 2FH, używanego do wkrywania zainstalowanego programu, a także do komunikacji między programami znajdującymi się w pamięci. Trzecia metoda polega na wykorzystaniu przerwania o numerach od 60h do 67h (ang. user interrupt). Przerwania tych można używać w programach bez obawy, że zniszczą któreś z ważnych przerwania systemu DOS. Oczywiście, wykorzystując któreś z tych przerwania, należy uprzednio sprawdzić, czy nie jest ono używane przez jakiś inny program.

### **Moment wywołania programu**

Kolejne problemy są związane z tym, że program rezydentny nie zawsze może w dowolnym momencie przerywać wykonywanie aktualnie realizowanego programu. Niezbyt odpowiedni jest moment, w którym aktywny program korzysta z funkcji systemu operacyjnego DOS [1]. Jest to spowodowane tym, że funkcje DOS nie są rekurencyjne i DOS przy ponownym wywołaniu określonej funkcji nie zmienia stosu tej funkcji. Powoduje to, że drugie wywołanie korzysta ze stosu pierwszego wywołania, co na ogół nie jest dozwolone. Zatem wywołanie jakiegokolwiek funkcji DOS w programie rezydentnym w trakcie wykonywania tej funkcji spowoduje zniszczenie stosu tej funkcji, a co za tym idzie — zawieszenie się systemu.

Rozwiązanie tego problemu umożliwia struktura systemu DOS. Otóż w trakcie wykonywania dowolnej funkcji systemu DOS nadaje on pewnej wewnętrznej fladze wartość różną od zera. W przeciwnym przypadku wartość tej flagi równa jest zeru. Pozostaje zatem odczytać jedynie wartość tej flagi, aby określić, czy DOS jest aktualnie aktywny, czy też nie. Aby to sprawdzić, można skorzystać z funkcji, która zwraca adres wewnętrznej flagi DOS (funkcja numer 34H).

### **Pamięć dysku**

W momencie uaktywniania programu rezydentnego nie wolno przerwać wykonywania operacji dyskowych. Procedura uaktywniająca program rezydentny musi zatem sprawdzić, czy aktualnie wykonywany program nie zapisuje bądź nie czyta danych z dysku.

### **Stos**

Jest to pewien obszar pamięci o specyficznym trybie dostępu. Informacje można zapisać tylko na wierzchołku stosu i odczytać również tylko z wierzchołka. Stos jest przeznaczony



do przechowywania adresów powrotnych dla podprogramów oraz do przechowywania danych potrzebnych przede wszystkim do kontynuowania programów przerwanych. Informacje zapisane na stosie danego programu nie powinny być zmieniane przez inny program, gdyż najczęściej prowadzi to do błędnej pracy realizowanego programu. Podczas aktywizacji programów rezydentnych należy pamiętać o zachowaniu zawartości stosu przerwanych programów w pamięci.

### **Blok wstępny programu**

Każdemu programowi wprowadzonemu do pamięci operacyjnej system DOS przydziela pewien jej obszar. Początek tego obszaru, określony jako początek segmentu programu, ma istotne znaczenie w systemie DOS, gdyż w tym segmencie jest umieszczany blok wstępny programu. Blok ten zajmuje 256 bajtów i stanowi obszar komunikacji między programem a systemem. W tym bloku są umieszczone pewne informacje o aktualnym programie, takie jak np. adres powrotu po zakończeniu programu, adres pierwszej nie zajętej przez program komórki pamięci itd. W momencie uaktywnienia programu rezydentnego aktualnym blokiem wstępnym jest blok przerwanych programów i jak w przypadku stosu trzeba przechować jego wartość i ustawić właściwy blok dotyczący programu rezydentnego. Po zakończeniu trzeba odtworzyć zawartość bloku poprzednio realizowanego programu.

## **2. REZIDENTNY PROGRAM BAZY DANYCH**

### **2.1. Struktura i wyprowadzenie danych do programu**

Programy rezydentne można wykorzystywać do realizacji różnego rodzaju zadań. Można je stosować do obsługi bazy danych, w programach ekspertowych i w programach sterujących. W dalszej części artykułu omówiono przykładowe wykorzystanie programu rezydentnego do zarządzania bazą danych. Program tworzenia bazy danych nie jest programem rezydentnym. Jego zadaniem jest przyjmowanie danych od użytkownika i zapamiętanie ich w pliku w postaci pewnej struktury ułatwiającej wyszukiwanie. Można tworzyć bazy danych z różnych dziedzin, więc należy zdecydować, czy wszystkie informacje będą przechowywane w jednym pliku czy też dla każdej dziedziny tworzy się oddzielny plik. Zastosowanie rozwiązania polegającego na stworzenie odrębnego pliku dla każdej dziedziny prowadzi do spowolnienia pracy. Zatem ze względu na szybki dostęp do danych wybrano pierwsze rozwiązanie. Program może przyjmować dane z dowolnych dziedzin życia, zachodzi więc potrzeba, aby nie kolidowały one ze sobą. Poniżej opisano sposób wprowadzania danych spełniający powyższe wymagania.

Na początku tekstu jest wyprowadzony tytuł, który jest ciągiem składającym się z 10 znaków. Jest on podawany jeden raz na początku dla każdej dziedziny. Tytuł jest ciągiem

znaków, więc trzeba go odróżnić od innych informacji w tekście. Można to zrobić umieszczając na początku tekstu specjalne znaki, które będą informować program, że to, co następuje po nich, jest tytułem. Wszystkie informacje zapisane w pliku będą wówczas związane z podanym tytułem. Po podaniu tytułu wprowadza się właściwą informację. Musi ona być odpowiednio sformatowana, po to by łatwo było później ją wyszukiwać. Informacja jest odszukiwana na podstawie pewnych charakterystycznych hasel. Hasła te musi określić użytkownik podczas przygotowywania danych. Wyróżnia się tu hasła główne i hasła rozgałęzione, uszczegóławiające zakres informacji podawany w hasle głównym. Podobnie jak tytuły, także hasła trzeba odróżniać od reszty tekstu. Należy także odróżniać hasła główne od hasel rozgałęzionych, które są związane z danym hasłem głównym. Przed hasłem głównym trzeba podać słowa kluczowe, które wyróżniają zarazem to hasło od reszty tekstu, jak i od hasła rozgałęzionego. Nazwą hasła jest ciąg o długości do 20 znaków. Po nazwie hasła podaje się jego opis. Opisem może być dowolny tekst z wyraźnie oznaczonym początkiem i końcem. Przyjęto, że dla oznaczenia początku i końca opisu hasła będą służyć nawiasy klamrowe ({ }). Ponieważ zdarza się, że opis hasła jest tak obszerny, iż nie mieści się cały na ekranie — to należy podzielić taki opis na kilka mniejszych części i pokazywać je po kolei (jak gdyby na kolejnych stronach książki). Można także wyświetlić na ekranie tylko nazwy określające poszczególne mniejsze fragmenty dotyczące danego hasła, a użytkownik sam zdecyduje, który z tych fragmentów go interesuje i następnie wyświetlić ten właśnie fragment. Aby umożliwić tego typu postępowanie, wprowadzono pojęcie hasel rozgałęzionych. Hasłem rozgałęzionym może być zatem nazwa mniejszego fragmentu informacji dotyczących danego hasła głównego. Nazwa hasła rozgałęzionego musi wystąpić w opisie hasła głównego i musi być ona tam specjalnie wyróżniona. Dla każdego hasła rozgałęzionego należy podać także jego opis, czyli tekst, który jest wyświetlany po wybraniu tego hasła. Również w opisie hasel rozgałęzionych mogą wystąpić odwołania do kolejnych hasel rozgałęzionych. A zatem struktura hasel jest strukturą hierarchiczną, co oznacza, że każde hasło poziomu wyższego może odwoływać się do dowolnej liczby hasel poziomu niższego. Dozwolone są przy tym także bezpośrednie odwołania skrócone do hasel na tym samym poziomie lub hasel na poziomie niższym i wyższym, ale związanych z tym samym hasłem głównym.

### Przykład Wyprowadzanie danych

++Tytuł

\*YEMEN hasło\_główne1

{ opis hasła głównego ...

...

] hasło\_rozgałęzione11

] hasło\_rozgałęzione12

...

```

J haslo_rozgałezione1n
}
*SANAA haslo_rozgałezione11
{ opis hasła rozgałezionego
    ....
J haslo_rozgałezione111
    ....
}
*SANAA haslo_rozgałezione12
{ opis hasła rozgałezionego
    ....
J haslo_rozgałezione121 (ewentualnie, jeśli potrzebne)
    ....
}
{

```

Zastosowano następujące zasady wprowadzania danych:

- na początku pliku musi być podana nazwa tytułu wyróżniona znakiem (++) ,
- przed wprowadzeniem hasła głównego należy podać określony wyraz (w przykładzie \*YEMEN),
- początek opisu hasła głównego stanowi znak "{", a jego koniec — znak "}",
- nazwa hasła rozgałezionego podawana w tekście opisu hasła głównego jest wyróżniona znakiem o kodzie 245 (J); po nim musi nastąpić nazwa hasła rozgałezionego, natomiast jego końcem jest znak zmiany linii lub kropka (jeśli nie chce się zmieniać linii),
- nazwa hasła rozgałezionego jest podawana po wprowadzeniu wyrazu (w przykładzie \*SANAA), a jego nazwa musi być taka sama jak w opisie hasła głównego,
- kończąc strukturę hasła głównego, trzeba napisać znak o kodzie 244 (I).

## 2.2. Korzystanie z bazy danych

Program korzystania z bazy danych jest programem rezydentnym. Umożliwia on użytkownikowi przeglądanie informacji, które zostały uprzednio wprowadzone, wyświetlając je na ekranie, a także pozwala użytkownikowi na w miarę dowolne poruszanie się po ekranie i poszukiwanie interesującej go informacji [3]. Ponieważ opisywany program jest programem



rezydentnym, można go wywołać w dowolnym czasie i z dowolnego miejsca na ekranie. Wywołany program sam zapamiętuje aktualny stan ekranu w buforze — utworzy nowy własny ekran, a po wyjściu z programu przywraca poprzedni. Na nowym ekranie są wyświetlane wszystkie hasła główne związane z danym tytułem. Aby umożliwić użytkownikowi poruszanie się po ekranie, trzeba określić znaczenie poszczególnych klawiszy interpretowanych przez program. Program rozpoznaje naciśnięcie kilku klawiszy. I tak, naciskając dowolny z klawiszy sterujących kursorem ↓, ↑ czy też klawisz PgUp lub PgDn, można poruszać się po ekranie w odpowiednich kierunkach. Naciśnięcie klawisza F10 powoduje pokazanie u góry ekranu nowej strony, która zawiera wszystkie tytuły wprowadzone przez użytkownika i umożliwia mu wybranie jednego z nich. Aby zakończyć działanie programu, należy nacisnąć klawisz Esc. Program przywraca poprzednią zawartość pamięci ekranu przechowywaną dotychczas w buforze oraz przepisuje zawartość stosu, blok wstępny programu oraz stan funkcji DOS i BIOS, które zostały zapamiętane w momencie wywołania programu rezydentnego.

### 3. UWAGI KOŃCOWE

Doświadczenia zebrane w trakcie realizacji opisanego programu rezydentnego pozwalają na kilka uwag natury ogólnej. Problemy, które wystąpiły podczas budowy programu, są typowe dla różnych rozwiązań i obszarów zastosowań programów rezydentnych. Zaproponowane sposoby rozwiązania tych problemów mogą być wykorzystane również w innych zastosowaniach takich programów, jak przykładowo programy ekspertowe lub programy obsługi autonomicznych systemów automatyzacyjnych i pomiarowych.

Opisany rezydentny program korzystania z bazy danych ma szereg zalet i w wielu sytuacjach może stanowić wygodny sposób dostarczania dowolnych informacji związanych lub nie związanych z aktualnie realizowanym programem. Sposób organizacji danych pozwala na stosunkowo szybkie wyszukiwanie potrzebnych informacji bez konieczności zakończenia pracy aktualnie wykonywanego programu. Pewną niedogodność stanowi tu rezydentny charakter programu, z czym związany jest fakt, że program cały czas znajduje się w pamięci komputera ograniczając w pewnym stopniu zasoby pamięci dostępnej dla aktualnie realizowanego programu. Jednak duże zasoby pamięci operacyjnej współczesnych komputerów klasy IBM PC sprawiają, że ograniczenia te są mało istotne. Natomiast zalety programów rezydentnych skłaniają do poszukiwań tego rodzaju rozwiązań oprogramowania dla małych systemów pomiarowych służących do współpracy komputera z inteligentnymi przyrządami pomiarowymi pracującymi na bieżąco.

## LITERATURA

1. Kotowski M.: Pod zegarem. LUPUS, Warszawa 1992.
2. Boneler D.: DOS od środka. Oficyna Wydawnicza HELP, Warszawa. 1990
3. Al Raimi H.: Zaprojektować i wykonać zespół narzędzi programowych do wyświetlania testów informacyjnych z wykorzystaniem koncepcji "HYPERTEXT". Praca magisterska nr 969. Instytut Informatyki, Politechnika Śląska, Gliwice 1992.

Recenzent: Prof. dr hab. inż. Augustyn Chwaleba

Wpłynęło do Redakcji dnia 21 kwietnia 1995 r.

## Abstract

Resident programs can be the important part of IBM PC class computer software assisting its basic programs execution. It especially applies to resident programs servicing database and expert programs. They can be activated at any time by an operator which causes an instantaneous stoppage of the program executed at the moment. The advantage of such a solution is the simplicity of using resident programs.

Basic problems related to resident program design in DOS have been described in the paper. Problems typical for resident programs referring to stack use, program preliminary block service and ways of disk memory resources use have been discussed as well. The exemplary resident program servicing the simple database has been presented in the paper, too.