Ćwiczenie 5. Funkcje biosu, dos i programy rezydentne

Celem ćwiczenia jest napisanie prostego programu rezydentnego przy wykorzystaniu funkcji dostępnych w systemie operacyjnym.

Program w C do realizacji w asemblerze

```
//
#include "dos.h"

void interrupt (*old_1c)(void); //wskaznik "daleki"
int volatile i;

//zachowaj wszystkie rejestry na stosie
void interrupt new_1c(void)
{
i++;
//w tym miejscu wyswietl znak wyznaczony jako
// (i%10)+'0' w lewym gornym rogu ekranu
//1 - za pomoca funkcji BIOS-u (przerwanie int 10H*)
//2 - bezposrednio adresujac pamiec ekranu
old_1c();  //wywolanie starej funkcji obslugi przerwania
} //odtworz rejestry i powroc z przerwania

void main(void)
{
old_1c = getvect(0x1c); // pobierz wektor przerwania za
                        // pomoca funkcji systemu DOS
setvect(0x1c,new_1c);   //ustaw wektor
keep(0,8192/16);         //zakoncz program, zostawiając go rezydentnym
}
```

Wskazówka do programu

1. Do wypisania znaku na ekranie można wykorzystać funkcję BIOS-u nr 9 w ramach przerwania 10H. Funkcja ta wypisuje znak w pozycji kursora.

AH	09h
AL	znak do wyświetlenia (kod ASCII)
BH	numer strony (00h - numer strony pierwszej)
BL	atrybut (tryb tekstowy) lub kolor (tryb graficzny), jeśli 7 bit jest ustawiony w <256-kolorowym trybie graficznym, znak jest XOR'owany z obrazem
CX	ilość znaków - ile razy ma być powtórzony znak

Zatem funkcja wypisująca znak na ekranie może wyglądać następująco:

```
union REGS regs;
void napiszZnak( unsigned char strona,
             unsigned char znak, unsigned char atrybut, unsigned int ile )
{
regs.h.ah = 9;  // numer funkcji zawsze w AH
regs.h.al = znak;
regs.h.bh = strona;
regs.h.bl = atrybut;
regs.x.cx = ile;
int86( 0x10, &regs, &regs );
}
```

2. Drugi sposób polega na pisaniu bezpośrednio do pamięci ekranu. Pamięć ekranu w trybie tekstowym rozpoczyna się pod adresem B800:0000. Każdy znak jest opisany dwoma bajtamu (atrybut, kod ASCII). Zatem jeśli:

```
int far *ptr=MK_FP(0xB800,0); // wskaznik na poczatek pamieci obrazu
```

to:

```
ptr[0]=0x1F00|'A';            // wpisz znak bezposrednio do pamieci obrazu
```

spowoduje wypisanie znaku „A” na ekranie w kolorze tła 1 (ciemnoniebieski) i kolorze znaku 0xF (biały).

Przebieg ćwiczenia

- przygotować plik `lab5.asm`
- wykonać kompilację i linkowanie
- uruchomić `lab5.exe` z poziomu systemu DOS

Warunki zaliczenia ćwiczenia

Zaliczenie ćwiczenia polega na demonstracji działania programu prowadzącemu.

Sprawozdanie

Sprawozdanie powinno zawierać wydruk programu z komentarzami.

Literatura

Kernigham, Ritchie, *Język C*, WNT 1989
Dokumentacja pakietu Borland C 3.1
Wróbel Eugeniusz, *Asembler 8086/88*
Scanlon Leo J., *Assembler 8086/8088/80286*
Kruk Stanisław, *Język Assembler dla początkujących*
Syck Gary, *Turbo Assembler : biblia użytkownika*