

Marcin Stępnia

# Architektura systemów komputerowych

## Laboratorium 11

### Symulator SMS32

### Przerwania programowe

#### 1. Informacje

Przerwanie jest to sygnał do procesora wskazujący zdarzenie, które wymaga natychmiastowej uwagi. Pojawienie się przerwania powoduje wstrzymanie aktualnie wykonywanego programu i wykonanie przez procesor kodu procedury obsługi przerwania.

Typowe procedury obsługi przerwań obsługują naciśnięcia klawiszy, ruchy myszy, wypisywanie na ekran, czytanie i pisanie na dysku i tak dalej.

Przerwanie jest jak procedura, ale wywoływana w inny sposób. Procedury są wywoływane przez skok do adresu początku procedury. Ten adres jest znany tylko dla programu, który posiada tę procedurę. Przerwania są wywoływane przez wyszukanie adresu kodu przerwania w tablicy wektorów przerwań. Zawartość tej tablicy jest publikowana i powszechnie znana. Na przykład system MS DOS w znaczący sposób korzysta z przerwań dla wszystkich operacji związanych z dyskiem, ekranem, myszą, klawiaturą, siecią i innymi usługami.

Pisząc swój własny kod i ustawiając, aby wektor przerwania wskazywał do punktu we własnym kodzie, zachowanie przerwania może być całkowicie zmienione. Własny kod obsługujący przerwanie może dodać kilka przydatnych zachowań, a następnie skoczyć z powrotem do oryginalnego kodu, aby zakończyć pracę.

Przerwania dzielimy na:

- programowe - uruchamiane na żądanie, przez programy.
- sprzętowe - wywoływane za pomocą sygnałów elektronicznych do procesora z urządzeniami sprzętowymi.

W przypadku przerwań programowych, po napotkaniu odwołania do przerwania - procesor przerywa wykonywanie aktualnego programu i "przeskakuje" do procedury wywołanego przerwania.

### 1.1. Tablica wektorów przerwań

Wektory przerwań symulatora znajdują się w pamięci pod adresami 00..0F. Ponieważ na początku pamięci musi znaleźć się instrukcja, która przeskoczy tablicę wektorów, to pierwszą użyteczną komórką dla wektora jest [02]. Podany zakres jest jednak tylko umowny. Podanie adresu dowolnej komórki przy wywołaniu przerwania zadziała prawidłowo.

### 1.2. Wywołanie przerwania

Do wywołania przerwania służy instrukcja *INT numer\_przerwania*. Numer przerwania to w rzeczywistości adres wektora przerwania. Na przykład instrukcja *INT 02* spowoduje odczytanie przez procesor wartości komórki o adresie 02. Następnie wskaźnik instrukcji jest odkładany na stosie, a program kontynuuje pracę od instrukcji znajdującej się pod adresem wskazanym przez odczytaną wartość. Dalej wykonywany jest kod przerwania. Jego wykonanie kończy instrukcja *IRET*, która powoduje powrót z przerwania (wskaźnik instrukcji jest przywracany ze stosu). W zrozumieniu działania przerwań powinien pomóc program przedstawiony na listingu 1.

Listing 1. Program demonstrujący działanie przerwań programowych

```
JMP    Start ; skocze za wektor przerwan
DB 52 ; wektor na pozycji 02 wskazujacy adres 52
DB 72 ; wektor na pozycji 03 wskazujacy adres 72
Start:
INT    02 ; wykonuje przerwanie 02
INT    03 ; wykonuje przerwanie 03
JMP    Start
; -----

ORG    50
DB 84 ; tablica bajtow stanow swiatel
DB 30
; Tutaj rozpoczyna sie kod przerwania
MOV    AL,[50] ; kopiuje z pamieci pod adresem 50 do rejestru AL
MOV    BL,[51] ; kopiuje z pamieci pod adresem 51 do rejestru BL
MOV    [50],BL ; zapisuje wartosc rejestru BL pod adresem 50
MOV    [51],AL ; zapisuje wartosc rejestru AL pod adresem 51
OUT    01 ; wysyla dane do portu swiatel
IRET    ; powrot z procedury obslugi przerwania
```

```

; -----
ORG 70
DB FC ; tablica bajtów ze stanami wyświetlacza
DB DE
; Tutaj rozpoczyna się kod przerwania
MOV AL,[70] ; kopiuje z pamięci pod adresem 70 do rejestru AL
;XOR AL,22 ; można wykorzystać zamiast definiowania drugiego bajtu
MOV BL,[71] ; kopiuje z pamięci pod adresem 71 do rejestru BL
MOV [70],BL ; zapisuje wartość rejestru BL pod adresem 70
MOV [71],AL ; zapisuje wartość rejestru AL pod adresem 71
OUT 02 ; wysyła dane do wyświetlacza siedmiosegmentowego
IRET ; powrót z procedury obsługi przerwania
; -----
END

```

### 1.3. Dodatkowe informacje

1. [http://www.softwareforeducation.com/sms32v50/sms32v50\\_manual/200-10swint.htm](http://www.softwareforeducation.com/sms32v50/sms32v50_manual/200-10swint.htm)
2. [https://pl.wikipedia.org/wiki/Bufor\\_cykliczny](https://pl.wikipedia.org/wiki/Bufor_cykliczny)
3. [https://en.wikipedia.org/wiki/Circular\\_buffer](https://en.wikipedia.org/wiki/Circular_buffer)
4. [http://www.vias.org/javacourse/chap16\\_04.html](http://www.vias.org/javacourse/chap16_04.html)

## 2. Zadania

### 2.1. Zadanie 1

Napisać program, który posiada procedurę do dzielenia. Dzielną przekazywana jest w rejestrze AL, a dzielnik w rejestrze BL. Wynik dzielenia zapisywany będzie do rejestru AL. Jeżeli jako dzielnik zostanie przekazane 0, to program uruchamia przerwanie 0D. Procedura obsługi tego przerwania wyświetli czerwone światło na sygnalizacji drogowej i ustawi flagę „O” (overflow). Rejestr flag nie może być modyfikowany bezpośrednio, dlatego trzeba tego dokonać za pośrednictwem stosu (instrukcja POPF).

### 2.2. Zadanie 2

Napisać program zawierający procedurę obsługi przerwania o numerze 03, realizującą odczyt cyfry z klawiatury. W rejestrze CL zostanie zapisana wartość cyfry (należy przekształcić znak ASCII). Jeżeli wprowadzony znak z klawiatury nie będzie cyfrą, to w rejestrze CL powinna znaleźć się liczba -1. Wyjątkiem jest litera „Q” (mała lub wielka), dla której rejestr CL przyjmuje

wartość -2. Program działa w pętli, wywołując przerwanie i sygnalizując jego wynik za pomocą świateł drogowych, według poniższego schematu:

- wykonuje x mignięć zielonym światłem (x - wprowadzona cyfra),
- dla zwróconej wartości -1 zapala się czerwone światło.

Pętla (a także program) kończy się po napotkaniu liczby -2 w rejestrze CL po wykonaniu przerwania.

### **2.3. Zadanie 3**

Napisać program z procedurą obsługi przerwania 04. Przerwanie to odczytuje znak z klawiatury i umieszcza go w buforze cyklicznym w pamięci. Wielkość takiego bufora należy ustalić na 5 bajtów. Po naciśnięciu przycisku „Enter” zawartość bufora wyświetla się na wyświetlaczu VDU. Program wykonuje przerwanie w pętli nieskończonej. Bufor nigdy nie przepełnia się, bo poprzednio dodane dane zostają nadpisane. Jeżeli wskaźnik zapisu zrówna się ze wskaźnikiem odczytu, to wskaźnik odczytu zostaje przesunięty. W takiej sytuacji nieodczytane znaki zostają utracone. W buforze można zapisać do 5 znaków, ale standardowo zaimplementowany bufor cykliczny pozwoli na odczyt tylko 4 z nich.

### **2.4. Zadanie dodatkowe**

Napisać program zawierający przerwanie 05, którego zadaniem jest wstawienie odczytanego z klawiatury znaku na kolejną wolną pozycję wyświetlacza VDU. Po naciśnięciu przycisku „Enter” dodawanie rozpoczyna się od nowej linii, a naciśnięcie przycisku „Backspace” powoduje usunięcie ostatnio wprowadzonego znaku. Opcjonalnie można dodać wyświetlanie kursora.