

OPERACJE PLIKOWE

- ▣ Jak wiemy, wszystkich danych nie zmieścimy w pamięci. A nawet jeśli zmieścimy, to pozostaną tam tylko do najbliższego wyłączenia zasilania. Dlatego trzeba je zapisywać do pliku, a potem umieć je z tego pliku odczytać.

- ▣ W Asemblerze nie ma do tego celu żadnych instrukcji (jak i do innych operacji z tym związanych). By stworzyć plik trzeba użyć pewnych przerwań.
- ▣ DOS - przerwanie systemowe 21h
- ▣ Linux – przerwanie systemowe 80h

DOS

Tworzenie nowego pliku

- ▣ Do tworzenia pliku używamy przerwania 21H i funkcji 3CH. Przerwanie wykonuje się w Asemblerze za pomocą instrukcji INT, a funkcja jest to wartość rejestru AH.

Tworzenie nowego pliku

- ▣ Przed wywołaniem tego przerwania w rejestrze CX podajemy atrybuty naszego pliku (o atrybutach powiemy później), a w DS:DX nazwę, jaką ma mieć. Nazwa ma być zapisana w kodzie ASCII, tzn. musi być zakończona znakiem o kodzie 0, np. jeśli chcemy stworzyć plik o nazwie "PLIK.TXT" napiszemy tak:

Tworzenie nowego pliku

```
MOV AH,3CH  
MOV CX,0  
MOV DX,OFFSET Nazwa  
INT 21H  
...  
Nazwa DB "PLIK.TXT",0
```

Jeśli operacja zakończy się sukcesem, to w rejestrze AX znajdzie się numer dojścia do pliku. Jest on bardzo ważny. Identyfikuje on plik, do którego będą później zapisywane dane. Jeśli jednak operacja tworzenia pliku zakończy się niepowodzeniem to zostanie ustawiony znacznik C, a rejestr AX będzie posiadał kod powstałego błędu.

Otwieranie pliku

- ▣ By otworzyć plik, należy użyć funkcji **3DH** przerwania 21H. Jako parametry podajemy: w DS:DX adres do nazwy pliku (ASCIIZ); AL - tryb dostępu.

▣ Mamy do dyspozycji następujące tryby:

bity	wartość	opis
7	0	Procesy potomne dziedziczą dojście wraz z numerem
4..6	000	Każdy proces może otworzyć plik wielokrotnie z prawem do pisania i/lub czytania, ale tylko w tym trybie
4..6	001	Pełna wyłączość. Nie mogą istnieć żadne inne dojścia do pliku.
4..6	010	Wyłączość pisania, pozostałe dojścia mogą mieć tylko prawo do czytania, jednak nie mogą być otwarte w trybie 000
4..6	011	Wyłączość pisania, pozostałe dojścia mogą mieć prawo tylko do czytania i mogą być otwarte w trybie 000
4..6	100	Mogą istnieć inne dojścia z prawem do czytania i/lub pisania, jednak nie mogą być otwarte w trybie 000
0..3	000	Prawo do czytania
0..3	001	Prawo do pisania
0..3	010	Prawo do czytania i pisania

- ▣ Zazwyczaj używa się trzech trybów:
 - $AL=0$ - czytanie
 - $AL=1$ - pisanie
 - $AL=2$ - czytanie i pisanie

- ▣ Jeśli więc chcemy otworzyć istniejący plik o nazwie PLIK.TXT napiszemy:

```
MOV AH,3DH
MOV AL,0
MOV DX,OFFSET Nazwa
INT 21H
...
Nazwa DB "PLIK.TXT",0
```

- ▣ Podczas otwierania mogą wystąpić również błędy i wtedy znacznik C jest ustawiany, a AX posiada kod błędu. Jeśli operacja została wykonana poprawnie C=0, a AX zawiera numer dojścia.

Zapis do pliku

- ▣ Jeśli mamy już otwarty plik i chcemy w nim coś zapisać musimy użyć funkcji 40H przerwania 21H. W rejestrze BX podajemy numer dojścia (uzyskaliśmy go podczas otwierania pliku), w rejestrach DS:DX podajemy adres początku danych, które chcemy zapisać, a w rejestrze CX podajemy ilość bajtów do zapisania. Jeśli operacja zakończy się sukcesem znacznik C jest wyzerowany, a AX zawiera liczbę bajtów zapisanych. Jeśli podczas zapisu wystąpił błąd C=1, a AX zawiera kod błędu.

- ▣ Jeśli więc chcemy zapisać w naszym (otworzonym wcześniej) pliku napis "KURS ASSEMBLERA" napiszemy:

```
MOV AH,40H
```

```
MOV BX,Numer dojścia
```

```
MOV DX,OFFSET Napis
```

```
MOV CX,15
```

```
INT 21H
```

```
...
```

```
Napis DB „HELLO WORLD.”
```

Czytanie z pliku

- ▣ Do czytania z pliku służy funkcja 3FH przerwania 21H. W rejestrze BX podajemy również numer dojścia, CX ilość bajtów, które chcemy przeczytać, a w DS:DX adres bufora, gdzie mają zostać one zapamiętane. Jeśli operacja zakończy się sukcesem w rejestrze AX będziemy mieli ilość przeczytanych bajtów, a znacznik C=0. Ilość przeczytanych bajtów nie musi być równa ilość bajtów, które kazaliśmy przeczytać. Jeśli plik zawiera np. 30 bajtów danych, a my mu każemy przeczytać 50 bajtów, to AX będzie zawierało 30 bajtów.

Zamknięcie dojścia do pliku

- ▣ Każdy plik, który został wcześniej otwarty trzeba na koniec zamknąć. Do tego celu służy funkcja `3EH` przerwania `21H`. Podajemy w rejestrze `BX` numer dojścia do pliku, który chcemy zamknąć. Jeśli wystąpi jakiś błąd to znacznik `C` zostanie ustawiony, a `AX` zawiera kod powstałego błędu.

Zmiana wskaźnika w pliku

- ▣ Do ustawiania wskaźnika w pliku służy funkcja 42H. W rejestrze BX podajemy numer dojścia, w AL sposób przesunięcia wskaźnika, a w CX:DX odległość na jaką chcemy przesunąć. Istnieją trzy rodzaje przesunięć:
 - ▣ AL=0 - licząc od początku pliku
 - ▣ AL=1 - licząc od bieżącej pozycji kursora
 - ▣ AL=2 - licząc od końca zbioru

- ▣ Przesunięcie względem końca zbioru używa się również do obliczania wielkości zbioru, gdyż wynikiem funkcji 42H jest w DX:AX wartość aktualnego położenia (o ile nie wystąpił błąd, wtedy C=1, a AX kod błędu).

Usuwanie pliku

- ▣ Do usuwania zbiorów z dysku służy funkcja 41H przerwania 21H. W rejestrze DS:DX podajemy nazwę zbioru w kodzie ASCII. Jeśli nie wystąpi błąd znacznik C zostaje ustawiony, a AX zawiera kod powstałego błędu.

Sprawdzenie atrybutów pliku

- ▣ Jeśli chcemy sprawdzić jakie dane plik ma atrybuty musimy posłużyć się funkcją 43Hprzerwania 21H. Podajemy wówczas w AL wartość 0, a w DS:DX nazwę pliku w kodzie ASCII. Po wykonaniu przerwania w CX będą atrybuty pliku.

- ▣ Atrybuty mają następujący format:
 - bit0 - tylko do odczytu
 - bit1 - ukryty
 - bit2 - systemowy
 - bit5 - archiwalny

Nadawanie atrybutów pliku

- ▣ Do nadawania atrybutów dla pliku służy również funkcja 43H. Podajemy teraz w AL wartość 1, w DS:DX nazwę pliku, a w rejestrze CX atrybuty, które chcemy nałożyć.

Zmiana nazwy pliku

- ▣ Czasami zachodzi potrzeba zmiany nazwy pliku. Wtedy trzeba posłużyć się funkcją 56H. W rejestrze DS:DX podajemy pierwotną nazwę pliku, a w rejestrze ES:DI nową nazwę pliku. Jeśli operacja zakończy się niepowodzeniem zostanie ustawiony znacznik C, a rejestr AX będzie zawierał kod błędu.

Linux

Czytanie z pliku (sys_read)

- ▣ Numer funkcji – 3
- ▣ Jako argumenty funkcji podajemy:
 - EBX = deskryptor pliku
 - ECX = adres bufora docelowego
 - EDX = liczba bajtów do przeczytania
- ▣ Funkcja zwraca w EAX liczbę odczytanych bajtów lub kod błędu

Zapis do pliku (sys_write)

- ▣ Numer funkcji – 4
- ▣ Jako argumenty funkcji podajemy:
 - EBX = deskryptor pliku
 - ECX = adres bufora źródłowego
 - EDX = liczba bajtów do zapisania
- ▣ Funkcja zwraca w EAX liczbę zapisanych bajtów lub kod błędu

```
MOV EAX, 4           ; numer funkcji systemowej:
                     ; sys_write - zapisz do pliku
MOV EBX, 1           ; numer pliku, do którego pisze.
                     ; 1 = standardowe wyjście = ekran
MOV ECX, tekst       ; ECX = adres (offset) tekstu
MOV EDX, dlugosc     ; EDX = długość tekstu
INT 80h             ; wywołanie funkcji systemowej
```

...

```
section .data
```

```
tekst DB "Czesc", 0ah ; napis
dlugosc EQU $ - tekst ; długość napisu
```


Otwarcie pliku (sys_open)

- ▣ Numer funkcji – 5
- ▣ Jako argumenty funkcji podajemy:
 - EBX = adres nazwy pliku ASCIIZ
 - ECX = bity dostępu
 - EDX = prawa dostępu / tryb
- ▣ Funkcja zwraca w EAX deskryptor pliku lub kod błędu

```
MOV    EAX, 5           ; numer funkcji systemowej:
MOV    EBX, plik       ; adres nazwy pliku
MOV    ECX, 2          ; 2 - otwieranie do odczytu i zapisu
MOV    EDX, 111000000b ; 700 - zabroń innym dostępu
```

```
INT    80h             ; jeśli udało się otworzyć plik
...                ; w EAX znajdzie się deskryptor
                    ; pliku
```

```
section .data
```

```
plik db "nazwa.txt",0 ; nazwa pliku
```

Zamknięcie pliku (sys_close)

- ▣ Numer funkcji – 6
- ▣ Jako argumenty funkcji podajemy:
 - EBX = deskryptor pliku
- ▣ Funkcja zwraca w EAX „0” lub kod błędu

...

```
MOV EAX, 6  
systemowej:
```

; numer funkcji

```
MOV EBX, <deskryptor_pliku>  
pliku
```

; EBX = deskryptor

```
INT 80h
```

...

Utworzenie pliku (sys_creat)

- ▣ Numer funkcji – 8
- ▣ Jako argumenty funkcji podajemy:
 - EBX = adres nazwy pliku ASCIIZ
 - ECX = prawa dostępu / tryb
- ▣ Funkcja zwraca w EAX deskryptor pliku lub kod błędu

Usunięcie pliku (sys_unlink)

- ▣ Numer funkcji – 8
- ▣ Jako argumenty funkcji podajemy:
 - EBX = adres nazwy pliku ASCII
- ▣ Funkcja zwraca w EAX „0” lub kod błędu