

Pamięć i pętle.

Tadeusz Puźniakowski

PJWSTK

2016

Spis treści

- 1 Adresowanie
 - Tryby pracy procesora
- 2 Pętle
 - Pętla typu loop
- 3 Matematyka w Asm
 - Dodawanie i odejmowanie
 - Logika
 - Mnożenie i dzielenie

Mała uwaga

Jak zwykle cały wykład jest w kontekście procesora z rodziny x86

Tryby pracy procesora

Tryb rzeczywisty

Procesor działa tak jak 8086. Podstawowy tryb pracy procesora x86. Każdy PC uruchamia się w tym trybie. Można zaadresować maksymalnie 1MB pamięci.

Tryb chroniony

Tryb wprowadzony w procesorach 80286. Dodaje wiele funkcjonalności koniecznych dla systemu wielozadaniowego, oraz obsługę większej pamięci.

Segmentacja pamięci

Segment

Pamięć jest podzielona na segmenty, w 16bitowym systemie segmenty rozpoczynają się co 16 bajtów.

Rejestr segmentowy

Rejestr przeznaczony do określania segmentu. Na 8086 są to: CS, DS, ES, SS

Adresowanie w NASM

Odwołanie się do adresu

```
mov dx, txt
```

```
...
```

```
txt: db "Hello", 0x0a, 0x0d," $"
```

Adresowanie w NASM

Pobranie danych z pod adresu

```
mov dx, [txt]                ; 6548h
...
txt: db "Hello", 0x0a, 0x0d,"$"
```

Adresowanie w NASM

Pobranie danych z pod adresu

```
mov dl , [txt+5]          ; 10
...
txt: db "Hello", 0x0a, 0x0d," $"
```


Adresowanie w NASM

Zapisanie danych pod adres

```
mov bx, 123
mov ax, 0A000h
mov es, ax
mov byte [es:bx], 15
mov word [es:bx], 15 ; 16bitow
```

Adresowanie w NASM

Pamięć VGA

Dostęp do pamięci VGA jest dość prosty, jest ona w tej samej przestrzeni adresowej co zwykła pamięć. Segment trybu txt to 0b800h, natomiast graficznego to 0A000h.

Adresowanie w NASM

Przykład na żywo

Ustawienie znaku i koloru na ekranie

Spis treści

- 1 Adresowanie
 - Tryby pracy procesora
- 2 Pętle
 - Pętla typu loop
- 3 Matematyka w Asm
 - Dodawanie i odejmowanie
 - Logika
 - Mnożenie i dzielenie

Etykiety w NASM

Etykieta globalna

Zwykła etykieta, taka jakiej używaliśmy do tej pory praktycznie zawsze. Zwykłe słowo zaczynające się od litery.

Etykieta lokalna

Zaczyna się od znaku kropki. Jest powiązana z ostatnią etykietą globalną. Pozwala na wielokrotne użycie etykiety lokalnej o tej samej nazwie w różnych blokach kodu.

Pętla

Komenda LOOP

W języku assemblera x86 istnieje komenda LOOP. Działa ona w ten sposób, że zmniejsza licznik CX (albo ECX) o 1 i jeśli nie jest on 0, to dokonuje skoku do adresu.

Pętla

Przykład pętli

```
mov cx, 10  
.petla  
...  
loop .petla
```

Makro

Przykład makra

```
%include "makra.asm"
```

```
%macro   pisz 1
```

```
mov dx, %1
```

```
mov ah, 9
```

```
int 0x21
```

```
%endmacro
```

```
pisz tekst
```


Spis treści

- 1 Adresowanie
 - Tryby pracy procesora
- 2 Pętle
 - Pętla typu loop
- 3 Matematyka w Asm
 - Dodawanie i odejmowanie
 - Logika
 - Mnożenie i dzielenie

Dodawanie

ADD cel, źródło

Instrukcja ADD

Wykonuje operację:
 $\text{cel} = \text{cel} + \text{zrodlo}$

Przykład

ADD AX, BX ; $\text{AX} = \text{AX} + \text{BX}$

Odejmowanie

SUB cel, źródło

Instrukcja SUB

Wykonuje operację:
 $\text{cel} = \text{cel} - \text{zrodlo}$

Przykład

SUB AX, BX ; $\text{AX} = \text{AX} - \text{BX}$

Operatory logiczne

Ogólna zasada

Operatory dwuargumentowe działają podobnie do dodawania i odejmowania:

$\text{cel} = (\text{cel})\text{operator}(\text{zrodlo})$

Najpopularniejsze operatory

AND, OR, XOR, SHR, SHL, NOT

Mnożenie

MUL czynnik

8 bit

Drugi czynnik jest zawsze w AL. Podajemy jedynie pierwszy czynnik. Wynik jest w AX.

$$AX = (\text{czynnik}) \times AL$$

16 bit

Drugi czynnik jest zawsze w AX. Podajemy jedynie pierwszy czynnik. Wynik jest w parze rejestrów DX,AX.

$$DX, AX = (\text{czynnik}) \times AX$$

Mnożenie

16 bit - ściągawka

$DX, AX = (\text{czynnik}) \times AX$

Przykład 16 bit

```
MOV AX, 0FFF0h
```

```
MUL word [czynnik]
```

```
...
```

czynnik dw 10h

Wynikiem jest $AX=0FF00h$, oraz $DX=0Fh$, czyli $DX,AX = 0FFF00h$

Dzielenie

DIV dzielnik

8 bit

Dzielna jest zawsze w rejestrze AX. Podajemy dzielnik. Wynik zawsze łąduje w AL, natomiast reszta w AH.

16 bit

Dzielna jest w parze rejestrów DX,AX. Dzielnik podajemy, natomiast wynik dzielenia jest w rejestrze AX. Reszta łąduje w DX.

$$AX = (DX, AX) / (\text{dzielnik})$$

$$DX = (DX, AX) \bmod (\text{dzielnik})$$

Dzielenie

16 bit - ściągawka

$$AX = (DX, AX) / (\text{dzielnik})$$
$$DX = (DX, AX) \bmod (\text{dzielnik})$$

Przykład 16 bit

```
MOV AX, 17
```

```
MOV DX, 0
```

```
DIV word [dzielnik]
```

```
...
```

```
dzielnik dw 8
```

Wynikiem jest AX=2, oraz DX=1

Kreska na ekranie

Przykład na żywo

Wyświetlanie kreski na ekranie.

Źródła

- intel.com
- wikipedia.org
- Moja wiedza