



Лекція 5

Арифметичні операції

професор кафедри комп'ютерних наук
Шило Галина Миколаївна



Системні виклики Linux

- 1) Задаємо номер системного виклику у регістрі EAX
- 2) Зберігаємо аргументи системного виклику у регістрах EBX, ECX
- 3) Викликаємо переривання 080h (0x80)
- 4) Результат повертается у регістр EAX

EAX	Назва	EBX	ECX	EDX
1	sys_exit	int		
3	sys_read	unsigned int	char *	size_t
4	sys_write	unsigned int	const char *	Size_t



Режими адресації

1) Регістрова адресація

```
mov eax, ebx
```

2) Пряма (або безпосередня) адресація

```
mov eax, 45h
```

```
add eax, 0Ah
```

```
value DB 15h;
```

```
add value, 23h;
```

3) Адресація пам'яті



Режими адресації

3) Адресація пам'яті

```
value_table DB 11h, 15h, 22h
```

```
mov CL, value_table[2] ; розміщуємо у регістрі 3-й елемент
```

```
mov CL, value_table+1 ; розміщуємо у регістрі 2-й елемент
```

```
value_table1 DW 1234h, 0011h, 1356h
```

```
mov CX, value_table1[1]
```

```
mov CX, value1_table1+1
```



Режими адресації

3) Непряма адресація пам'яті
value_table DB 11h, 15h, 22h
mov EBX, [value_table]
mov [EBX], 12h

num1 resb 2



Робота з числами з вхідного потоку

- 1) Отримати ASCII-код числа (однорозрядне)
- 2) Перевести ASCII-код до числа (вчитаємо 48 або 30h або '0')
- 3) Виконуємо арифметичну операцію
- 4) Перевести результат у вигляді цілого числа до ASCII-коду (додаємо 48 або 30h або '0')
- 5) Виводимо у вихідний потік



Резервування пам'яті

```
segment .data
```

```
num2 db 13h ; виділяємо байт для num2 та ініціалізуємо числом 13h
```

```
num3 dw 1234h; виділяємо 2 бати
```

```
segment .bss
```

```
num1 resb 2 ; для змінної зарезервували 2 байта
```

```
result resb 1 ; для змінної result зарезервували 1 байт
```



Ініціалізовані змінні

Директива	Розмір простору пам'яті
DB	1 байт, byte
DW	2 байта, word
DD	4 байта, Doubleword
DQ	8 байт, Quadword
DT	10 байт



Константи

Директива EQU

Ім'я константи equ вираз
num1 equ 50



Арифметичні операції

ADD перший операнд, другий операнд
SUB операнд1, операнд2

- 1) Регістр, регістр
- 2) пам'ять, регістр
- 3) регістр, пам'ять
- 4) регістр, константа
- 5) пам'ять, константа



Арифметичні операції

MUL винується для чисел без знаку
MUL множник

1) Якщо множення 2 значень типу byte

Операнд1 у регістрі AL x операнд2(множник) = AH+AL (AX)

2) Якщо множення 2 значень типу word

AX x 16-бітне значення = DX+AX

3) Якщо Doubleword

EAX x 32-бітне значення = EDX+EAX