

# Лабораторная работа №0

Игнатъев П. А.  
Б03-205

## Объявление глобальных переменных

```
.file "temp.c"
.text
.globl x
.data
.align 4
.type x, @object
.size x, 4
x:
.long 1
.globl y
.align 4
.type y, @object
.size y, 4
y:
.long 3
.globl z
.bss
.align 4
.type z, @object
.size z, 4
z:
.zero 4
.text
.globl main
.type main, @function
```

.globl – означает, что переменная глобальная;  
.align – размер ячейки памяти переменной  
.long (N) – означает, что переменная 4-х байтовое целое число, где N хранящееся в ней число;

## Простейший код и ассемблерный листинг

```
int x = 1, y = 3, z;
int main(){
    z = x + y;
    return 0;
}
```

```
movl    x(%rip), %edx
movl    y(%rip), %eax
addl    %edx, %eax
movl    %eax, z(%rip)
movl    $0, %eax
```

movl – присвоение регистру справа значения регистра слева

addl – сложение левого регистра с правым с последующим присвоением суммы правому регистру

subl – вычитание левого регистра из правого с последующим присвоением разности правому регистру

Для переменных различного типа данные команды будут отличаться  
Например:

- Для переменных типа double добавляется окончание **sd** (movsd)
- Для переменных типа float добавляется окончание **ss** (movss)
- Для переменных типа int добавляется окончание **l** (movl)

## Умножение и деление чисел

```
int x = 1, y = 3, z;  
int main(){  
    z = y * x;  
    return 0;  
}
```

```
movl    x(%rip), %eax  
imull   %edx, %eax  
movl    %eax, z(%rip)
```

Умножение целых чисел (**int**) происходит при помощи **imull**, результат умножения присваивается регистру справа.

Для **float** и **double** используется **mul** с соответствующими окончаниями описанными ранее.

**neg** – изменение знака на противоположный.

```
int x = 2, y = 4, z;  
int main(){  
    z = y / x;  
    return 0;  
}
```

```
movl    y(%rip), %eax  
movl    x(%rip), %ecx  
cld  
idivl   %ecx  
movl    %eax, z(%rip)
```

Деление осуществляет **div**

```
movsd   y(%rip), %xmm0  
movsd   x(%rip), %xmm1  
divsd   %xmm1, %xmm0  
movsd   %xmm0, z(%rip)
```

## Сравнение “Hello world!” в С и С++:

С

printf(“Hello world!”);

```
.LC0:  
    .string "Hello world!"  
    .text  
    .globl main  
    .type main, @function  
main:  
.LFB0:  
    .cfi_startproc  
    endbr64  
    pushq %rbp  
    .cfi_def_cfa_offset 16  
    .cfi_offset 6, -16  
    movq %rsp, %rbp  
    .cfi_def_cfa_register 6  
    leaq .LC0(%rip), %rax  
    movq %rax, %rdi  
    movl $0, %eax  
    call printf@PLT  
    movl $0, %eax  
    popq %rbp  
    .cfi_def_cfa 7, 8  
    ret  
    .cfi_endproc  
.LFE0:
```

С++

std::cout << “Hello world!”;

```
.LC0:  
    .string "Hello world!"  
    .text  
    .globl main  
    .type main, @function  
main:  
.LFB1731:  
    .cfi_startproc  
    endbr64  
    pushq %rbp  
    .cfi_def_cfa_offset 16  
    .cfi_offset 6, -16  
    movq %rsp, %rbp  
    .cfi_def_cfa_register 6  
    leaq .LC0(%rip), %rax  
    movq %rax, %rsi  
    leaq _ZSt4cout(%rip), %rax  
    movq %rax, %rdi  
    call _ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_Pkc@PLT  
    movl $0, %eax  
    popq %rbp  
    .cfi_def_cfa 7, 8  
    ret  
    .cfi_endproc
```

В целом отличается только способ вызова вывода.

Виды регистров общего назначения:

## Регистры общего назначения

- операнды логических и арифметических операций
- компоненты адреса
- указатели на ячейки памяти

- Аккумулятор
- Регистр данных
- Регистр-счетчик
- Базовый регистр
- Регистр указателя базы кадра стека
- Индекс источника
- Индекс приемника
- Регистр указателя стека

Регистры общего назначения  
целочисленного устройства

eax		ax	
		ah	al
edx		dx	
		dh	dl
ecx		cx	
		ch	cl
ebx		bx	
		bh	bl
	31	15	7 0
ebp		bp	
esi		si	
edi		di	
esp		sp	
	31	15	0