

## Лабораторная работа №8

Команда №8 (Смирнов Д. М., Бабич К. А.) - ПИН-21

**Задание Л8.31.** Разработайте программу, выводящую на стандартный вывод группу, номер и состав команды при помощи функции *puts()* библиотеки *libc* (аналогично заданию Л4).

**Результаты:**

```
.data
msg:
.string "PIN-21, Team 8, Smirnov D. M., Babich K. A.\n".text
.globl main
main:
sub $8, %rsp
lea msg(%rip), %rcx
sub $32, %rsp
call printf
add $32, %rsp
add $8, %rsp
xor %eax, %eax
ret
```

```
PIN-21, Team 8, Smirnov D. M., Babich K. A.
```

**Задание Л8.32.** Разработайте программу, реализующую ввод и последующий вывод (эхо) двух значений *x* и *y* следующих типов: – 16-битное целое; – 32-битное целое; – 64-битное целое; – 32-битное число с плавающей запятой; – 64-битное число с плавающей запятой с использованием библиотеки *libc* (в частности, функций *puts()/printf()/scanf()*).

Примечание: обратите внимание, что как *printf()*, так и *scanf()* имеют переменное число аргументов, что во многих соглашениях требует дополнительных действий при передаче параметров.

**Результаты:**

```
.data
scan_short:
.string "%hd %hd"
print_short:
.string "%hd, %hd"
scan_int:
.string "%d %d"
print_int:
.string "%d, %d"
scan_long:
.string "%ld %ld"
print_long:
.string "%ld, %ld"
scan_float:
.string "%f %f"
print_float:
.string "%f, %f"
```

```
scan_double:  
.string "%lf %lf"  
print_double:  
.string "%lf, %lf"  
msg:  
.string "\n"  
.text  
  
.macro SCAN type p q  
lea scan_\type(%rip), %rcx  
lea \p(%rsp), %rdx  
lea \q(%rsp), %r8  
sub $32, %rsp  
call scanf  
add $32, %rsp  
.endm  
  
.macro MSG  
lea msg(%rip), %rcx  
sub $32, %rsp  
call puts  
add $32, %rsp  
.endm  
  
.macro scan_print type p q  
SCAN \type \p \q  
lea print_\type(%rip), %rcx  
mov \p(%rsp), %rdx  
mov \q(%rsp), %r8  
sub $32, %rsp  
call printf  
add $32, %rsp  
MSG  
.endm  
  
.globl main  
main:  
sub $8, %rsp  
sub $48, %rsp  
  
scan_print short, 2, 0  
scan_print int, 8, 4  
scan_print long, 20, 12  
  
SCAN float, 32, 28  
lea print_float(%rip), %rcx  
cvtss2sd 32(%rsp), %xmm1  
movq %xmm1, %rdx  
cvtss2sd 28(%rsp), %xmm2  
movq %xmm2, %r8  
sub $32, %rsp  
call printf
```

```
add $32, %rsp
MSG

lea scan_double(%rip), %rcx
lea 44(%rsp), %rdx
lea 36(%rsp), %r8
sub $32, %rsp
call scanf
add $32, %rsp

lea print_double(%rip), %rcx
movq 44(%rsp), %rdx
movq 36(%rsp), %r8
sub $32, %rsp
call printf
add $32, %rsp

lea msg(%rip), %rcx
sub $32, %rsp
call puts
add $32, %rsp

add $48, %rsp
add $8, %rsp
xor %eax, %eax
ret
```

```
2
3 Sh
2, 3

123
1234 int
123, 1234

45678
34254326 long
45678, 34254326

123.3213
543.33 F1
123.321297, 543.330017

2.325345334
2135432.23 d
2.325345, 2135432.230000
```

**Задание Л8.33.** Разработайте программу, вычисляющую (вызывая функции libc/libm) по введённым значениям  $x$  и  $y$  с плавающей запятой двойной точности значение  $z$  (таблица Л8.1):

Вариант
$z = \text{pow}(x, y), \quad x^y$

**Результаты:**

```
.data
scan:
.string "%lf %lf"
print:
.string "%lf"
.text
.globl main
main:
sub $8, %rsp
sub $16, %rsp

lea scan(%rip), %rcx
lea 0(%rsp), %rdx
lea 8(%rsp), %r8
sub $32, %rsp
call scanf
add $32, %rsp

movq 0(%rsp), %xmm0
movq 8(%rsp), %xmm1
sub $32, %rsp
call pow
add $32, %rsp

lea print(%rip), %rcx
movq %xmm0, %rdx
sub $32, %rsp
call printf
add $32, %rsp

add $16, %rsp
add $8, %rsp
xor %eax, %eax
ret
```

```
C:\Qt\Tools\Qt
2
2
4.000000
```

**Задание Л8.34.** Задайте с клавиатуры  $N$  и  $x_0$  и напечатайте первые  $N$  членов целочисленной последовательности:

$$x_{i+1} = \begin{cases} x_i/2, & x_i \% 2 = 0 \\ 3x_i + 1, & x_i \% 2 \neq 0 \end{cases}$$

Для каждого из заданий указывайте ОС, разрядность программы и соответствующее им соглашение.

**Результаты:**

```
.data
scan:
.string "%llu %u"
print:
.string "=%llu\n"
.text

.globl main
main:
sub $8, %rsp
sub $16, %rsp

lea scan(%rip), %rcx
lea 0(%rsp), %rdx
lea 8(%rsp), %r8
sub $32, %rsp
call scanf
add $32, %rsp

mov $3, %r8
mov 0(%rsp), %rax

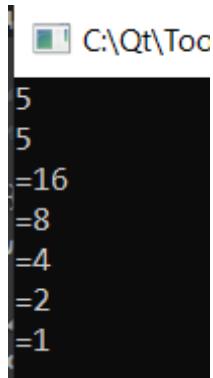
cycle:

bt $0, %rax
jc mark
shr $1, %rax
jmp endif
mark:
mul %r8
inc %rax
endif:

lea print(%rip), %rcx
push %rax
mov %rax, %rdx
sub $32, %rsp
call printf
add $32, %rsp
pop %rax
```

```
decl 8(%rsp)
jnz cycle

add $16, %rsp
add $8, %rsp
xor %eax, %eax
ret
```



The image shows a screenshot of a terminal window. The title bar reads "C:\Qt\Toc". The window contains the following assembly code:

```
5
5
=16
=8
=4
=2
=1
```