федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Московский институт электронной техники»

Направление подготовки 09.03.04 «Программная инженерия»

Дисциплина «Электротехника»

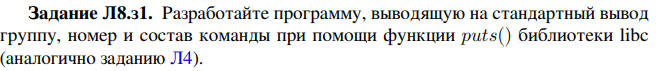
**Лабораторная РАБОТА №8**

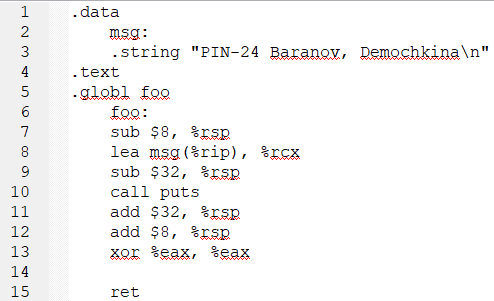
**Модули и функции. Вызов функций libc и libm.**

Работу выполнил студенты группы ПИН-24 Баранов Д.А. и Демочкина А.В.

Работу проверил ассистент Института СПИНТех Фомин Р.А.

**Цель работы:** изучить процесс компиляции программы на C++; научиться включать в проекты на языке C++ ассемблерные модули; изучить стандартные соглашения о вызовах и их соответствие платформам.





.data

msg:

.string "PIN-24 Baranov, Demochkina\n"

.text

.globl foo

foo:

sub $8, %rsp

lea msg(%rip), %rcx

sub $32, %rsp

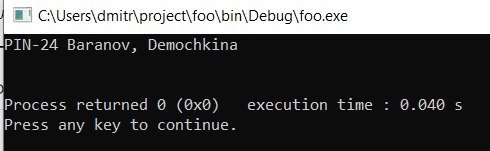
call puts

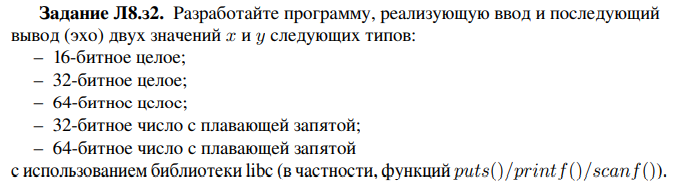
add $32, %rsp

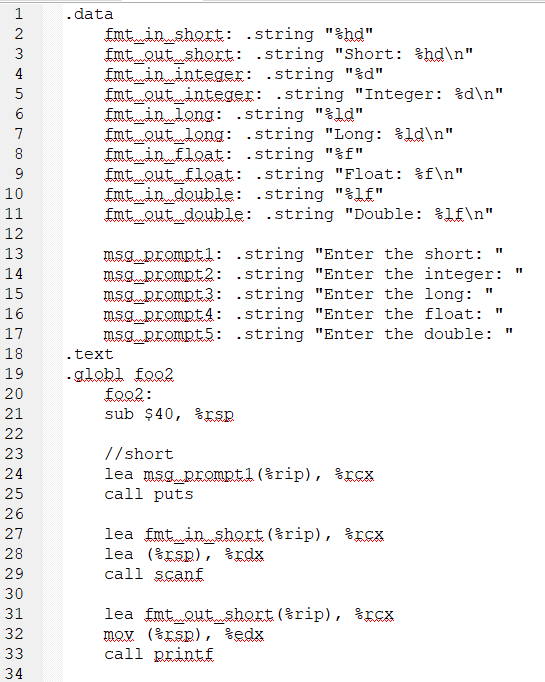
add $8, %rsp

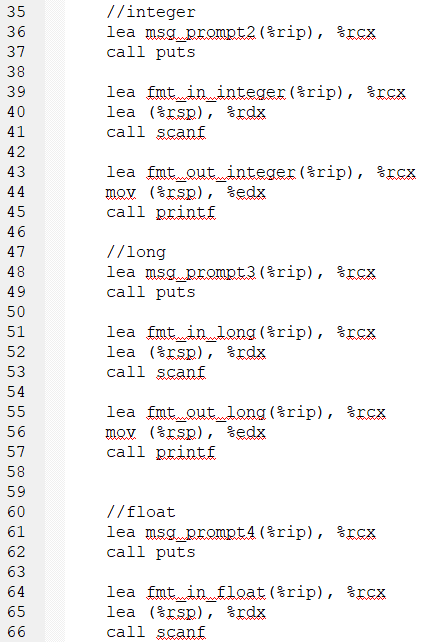
xor %eax, %eax

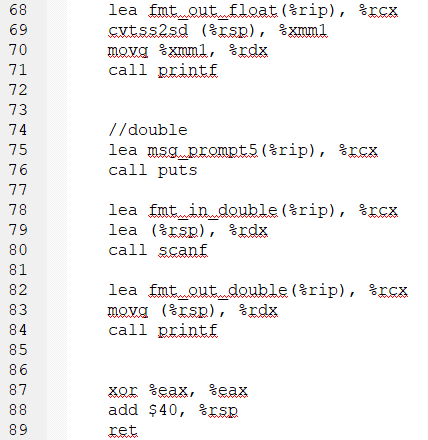
ret











.data

fmt\_in\_short: .string "%hd"

fmt\_out\_short: .string "Short: %hd\n"

fmt\_in\_integer: .string "%d"

fmt\_out\_integer: .string "Integer: %d\n"

fmt\_in\_long: .string "%ld"

fmt\_out\_long: .string "Long: %ld\n"

fmt\_in\_float: .string "%f"

fmt\_out\_float: .string "Float: %f\n"

fmt\_in\_double: .string "%lf"

fmt\_out\_double: .string "Double: %lf\n"

msg\_prompt1: .string "Enter the short: "

msg\_prompt2: .string "Enter the integer: "

msg\_prompt3: .string "Enter the long: "

msg\_prompt4: .string "Enter the float: "

msg\_prompt5: .string "Enter the double: "

.text

.globl foo2

foo2:

sub $40, %rsp

//short

lea msg\_prompt1(%rip), %rcx

call puts

lea fmt\_in\_short(%rip), %rcx

lea (%rsp), %rdx

call scanf

lea fmt\_out\_short(%rip), %rcx

mov (%rsp), %edx

call printf

//integer

lea msg\_prompt2(%rip), %rcx

call puts

lea fmt\_in\_integer(%rip), %rcx

lea (%rsp), %rdx

call scanf

lea fmt\_out\_integer(%rip), %rcx

mov (%rsp), %edx

call printf

//long

lea msg\_prompt3(%rip), %rcx

call puts

lea fmt\_in\_long(%rip), %rcx

lea (%rsp), %rdx

call scanf

lea fmt\_out\_long(%rip), %rcx

mov (%rsp), %edx

call printf

//float

lea msg\_prompt4(%rip), %rcx

call puts

lea fmt\_in\_float(%rip), %rcx

lea (%rsp), %rdx

call scanf

lea fmt\_out\_float(%rip), %rcx

cvtss2sd (%rsp), %xmm1

movq %xmm1, %rdx

call printf

//double

lea msg\_prompt5(%rip), %rcx

call puts

lea fmt\_in\_double(%rip), %rcx

lea (%rsp), %rdx

call scanf

lea fmt\_out\_double(%rip), %rcx

movq (%rsp), %rdx

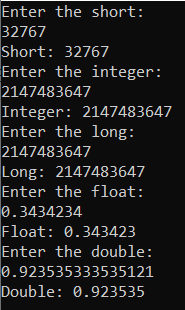
call printf

xor %eax, %eax

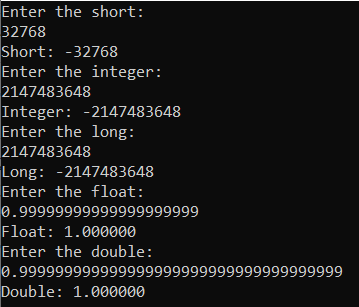
add $40, %rsp

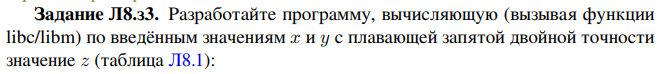
ret

**Результат:**



**Переполнение:**

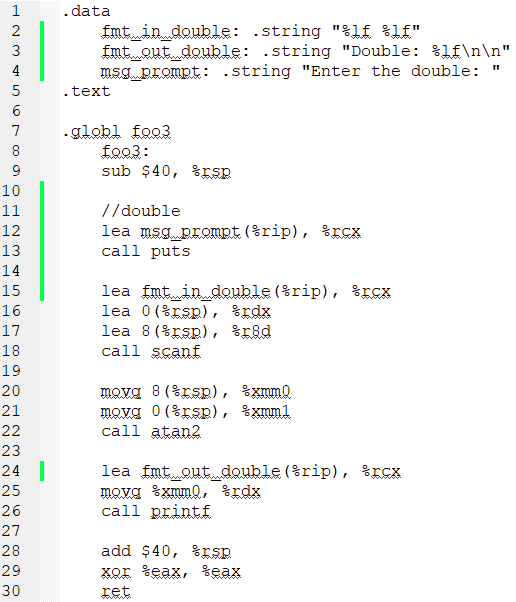




**Вариант 2.**



**Первый вариант:**



.data

fmt\_in\_double: .string "%lf %lf"

fmt\_out\_double: .string "Double: %lf\n\n"

msg\_prompt: .string "Enter the double: "

.text

.globl foo3

foo3:

sub $40, %rsp

//double

lea msg\_prompt(%rip), %rcx

call puts

lea fmt\_in\_double(%rip), %rcx

lea 0(%rsp), %rdx

lea 8(%rsp), %r8d

call scanf

movq 8(%rsp), %xmm0

movq 0(%rsp), %xmm1

call atan2

lea fmt\_out\_double(%rip), %rcx

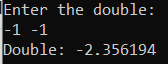
movq %xmm0, %rdx

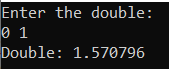
call printf

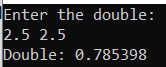
add $40, %rsp

xor %eax, %eax

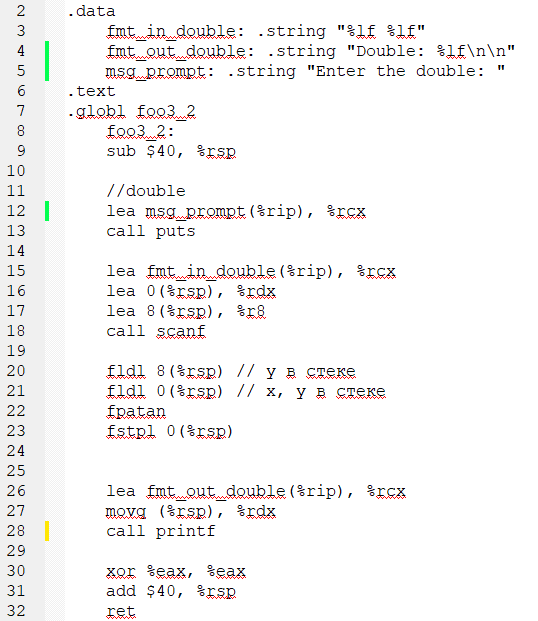
ret







**Второй вариант:**

****

.data

fmt\_in\_double: .string "%lf %lf"

fmt\_out\_double: .string "Double: %lf\n\n"

msg\_prompt: .string "Enter the double: "

.text

.globl foo3\_2

foo3\_2:

sub $40, %rsp

//double

lea msg\_prompt(%rip), %rcx

call puts

lea fmt\_in\_double(%rip), %rcx

lea 0(%rsp), %rdx

lea 8(%rsp), %r8

call scanf

fldl 8(%rsp) // y в стеке

fldl 0(%rsp) // x, y в стеке

fpatan

fstpl 0(%rsp)

lea fmt\_out\_double(%rip), %rcx

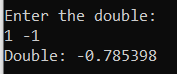
movq (%rsp), %rdx

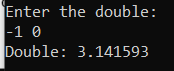
call printf

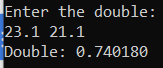
xor %eax, %eax

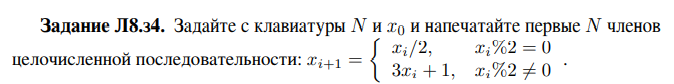
add $40, %rsp

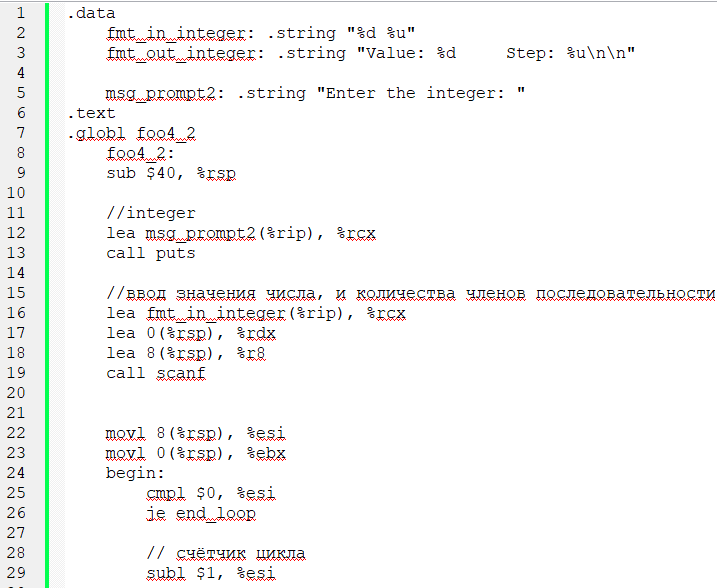
ret

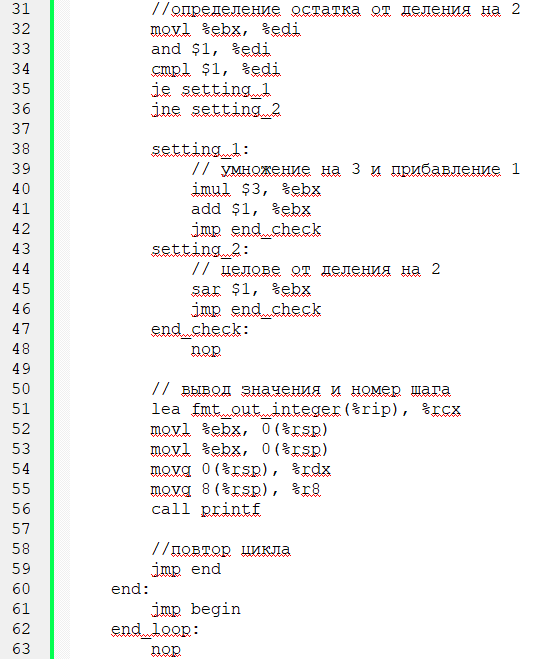














.data

fmt\_in\_integer: .string "%d %u"

fmt\_out\_integer: .string "Value: %d Step: %u\n\n"

msg\_prompt2: .string "Enter the integer: "

.text

.globl foo4\_2

foo4\_2:

sub $40, %rsp

//integer

lea msg\_prompt2(%rip), %rcx

call puts

//ввод значения числа, и количества членов последовательности

lea fmt\_in\_integer(%rip), %rcx

lea 0(%rsp), %rdx

lea 8(%rsp), %r8

call scanf

movl 8(%rsp), %esi

movl 0(%rsp), %ebx

begin:

cmpl $0, %esi

je end\_loop

// счётчик цикла

subl $1, %esi

//определение остатка от деления на 2

movl %ebx, %edi

and $1, %edi

cmpl $1, %edi

je setting\_1

jne setting\_2

setting\_1:

// умножение на 3 и прибавление 1

imul $3, %ebx

add $1, %ebx

jmp end\_check

setting\_2:

// целове от деления на 2

sar $1, %ebx

jmp end\_check

end\_check:

nop

// вывод значения и номер шага

lea fmt\_out\_integer(%rip), %rcx

movl %ebx, 0(%rsp)

movl %ebx, 0(%rsp)

movq 0(%rsp), %rdx

movq 8(%rsp), %r8

call printf

//повтор цикла

jmp end

end:

jmp begin

end\_loop:

nop

xor %eax, %eax

add $40, %rsp

ret

