

Лабораторная работа №8

Дисциплина: Архитектура компьютера

Комягин Андрей Николаевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Самостоятельная работа	10
3	Вывод	12

Список иллюстраций

2.1	Создание каталога	6
2.2	Работа программы 8.1	6
2.3	Работа программы 8.1.2	7
2.4	Работа программы 8.2	7
2.5	Работа программы 8.3	8
2.6	Вычисление произведения	9
2.7	Задание 1	11

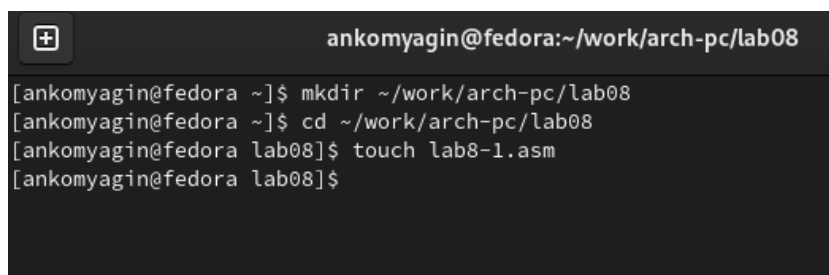
Список таблиц

1 Цель работы

Приобрести навыки написания программ с использованием циклов и обработкой аргументов командной строки.

2 Выполнение лабораторной работы

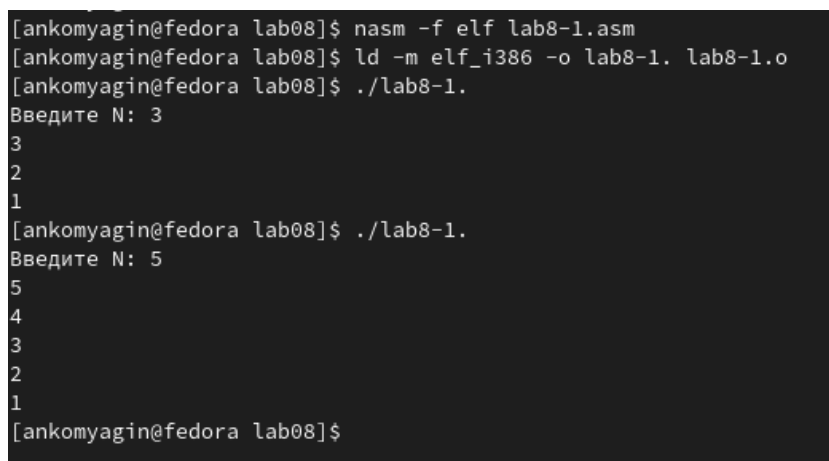
Создадим каталог для программ лабораторной работы 8. Создадим в нем файл **lab8-1.asm**(рис. 2.1).



```
ankomyagin@fedora:~/work/arch-pc/lab08
[ankomyagin@fedora ~]$ mkdir ~/work/arch-pc/lab08
[ankomyagin@fedora ~]$ cd ~/work/arch-pc/lab08
[ankomyagin@fedora lab08]$ touch lab8-1.asm
[ankomyagin@fedora lab08]$
```

Рис. 2.1: Создание каталога

Заполним файл в соответствии с листингом **8.1**. Создадим исполняемый файл и запустим его (рис. 2.2).



```
[ankomyagin@fedora lab08]$ nasm -f elf lab8-1.asm
[ankomyagin@fedora lab08]$ ld -m elf_i386 -o lab8-1. lab8-1.o
[ankomyagin@fedora lab08]$ ./lab8-1.
Введите N: 3
3
2
1
[ankomyagin@fedora lab08]$ ./lab8-1.
Введите N: 5
5
4
3
2
1
[ankomyagin@fedora lab08]$
```

Рис. 2.2: Работа программы 8.1

Изменим текст программы. Результат работы программы является некорректным. Снова изменим код, добавив команды **push** и **pop**, в данном случае число проходов соответствует N (рис. 2.3).

```
[ankomyagin@fedora lab08]$ nasm -f elf lab8-1.asm
[ankomyagin@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[ankomyagin@fedora lab08]$ ./lab8-1
Введите N: 4
3
2
1
0
[ankomyagin@fedora lab08]$
```

Рис. 2.3: Работа программы 8.1.2

Создадим новый файл. Заполним его в соответствии с листингом 8.2 и проверим работу (рис. 2.4). Программа обработала 4 аргумента (из-за 3-х пробелов)

```
ankomyagin@fedora:~/work/arch-pc/lab08
[ankomyagin@fedora ~]$ cd ~/work/arch-pc/lab08
[ankomyagin@fedora lab08]$ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
[ankomyagin@fedora lab08]$ ./lab8-2 1 5 3
1
5
3
[ankomyagin@fedora lab08]$
```

Рис. 2.4: Работа программы 8.2

Создадим новый файл. Заполним его в соответствии с листингом 8.3 и проверим работу (рис. 2.5).

```
2
[ankomyagin@fedora lab08]$ touch lab8-3.asm
[ankomyagin@fedora lab08]$ nasm -f elf lab8-3.asm
[ankomyagin@fedora lab08]$ ld -m elf_i386 -o lab8-3 lab8-3.o
[ankomyagin@fedora lab08]$ ./lab8-3
Результат: 0
[ankomyagin@fedora lab08]$ ./lab8-3 45 15 5 10
Результат: 75
[ankomyagin@fedora lab08]$
```

Рис. 2.5: Работа программы 8.3

Изменим текст программы, для вычисления произведения аргументов (рис. 2.6).

The image shows a terminal window and an IDE. The terminal window at the top displays the execution of an assembly program. The IDE below shows the source code for 'lab8-3.asm'.

```
[ankomyagin@fedora lab08]$ nasm -f elf lab8-3.asm
[ankomyagin@fedora lab08]$ ld -m elf_i386 -o lab8-3 lab8-3.o
[ankomyagin@fedora lab08]$ ./lab8-3 3 4 5
Результат: 0
[ankomyagin@fedora lab08]$ nasm -f elf lab8-3.asm
[ankomyagin@fedora lab08]$ ld -m elf_i386 -o lab8-3 lab8-3.o
[ankomyagin@fedora lab08]$ ./lab8-3 3 3 4 5
Результат: 60
[ankomyagin@fedora lab08]$ ./lab8-3 2 5 5
Результат: 50
[ankomyagin@fedora lab08]$
```

The IDE shows the following assembly code:

```
%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
    pop ecx ; Извлекаем из стека в `ecx` количество
    pop edx
    sub ecx,1
    mov esi, 1
next:
    cmp ecx,0h
    jz _end
    pop eax
    call atoi
    mov ebx,0
    add ebx,esi
    mul ebx
    mov esi,eax
    ; след. аргумент
    loop next
_end:
    mov eax, msg
    call sprint
    mov eax, esi
    call iprintLF
    call quit
```

Рис. 2.6: Вычисление произведения

2.1 Самостоятельная работа

Вариант 7

1. Напишем программу для нахождения суммы значений функции. Создадим программу и проверим её работу (рис. 2.7).

```
ankomyagin@fedora:~/work/arch-pc/lab08
[ankomyagin@fedora ~]$ cd ~/work/arch-pc/lab08

Открыть ▾ + • task_1.asm ~/work/arch-pc/lab08
lab8-1.asm | lab8-2.asm | lab8-3.asm | ● task_1.asm x

%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
msg1 db "Функция f(x) = 3(x+2): "
SECTION .text
global _start
_start:
mov eax,msg1
call sprintf
pop ecx
pop edx
sub ecx,1
mov esi, 0
next:
cmp ecx,0h
jz _end
pop eax
call atoi
add eax,2
mov ebx,3
mul ebx
add esi,eax
loop next
_end:
mov eax, msg
call sprintf
mov eax, esi
call iprintLF
call quit

[ankomyagin@fedora lab08]$ nasm -f elf task_1.asm
[ankomyagin@fedora lab08]$ ld -m elf_i386 -o task_1 task_1.o
[ankomyagin@fedora lab08]$ ./task_1 1 2 3
Функция f(x) = 3(x+2):
Результат: 36
[ankomyagin@fedora lab08]$ ./task_1 1 2
Функция f(x) = 3(x+2):
Результат: 21
[ankomyagin@fedora lab08]$ ./task_1 1 3 4
Функция f(x) = 3(x+2):
Результат: 42
[ankomyagin@fedora lab08]$
```

Рис. 2.7: Задание 1

3 Вывод

В ходе работы я приобрел навыки написания программ с использованием циклов и обработкой аргументов командной строки.