Лабораторная работа №8. Программирование цикла. Обработка аргументов командной строки.

Аджигалиева Амина Руслановна

Содержание

Список иллюстраций

1 Цель работы

Приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

2 Выполнение лабораторной работы

2.1 Реализация циклов в NASM

Создадим каталог для программам лабораторной работы № 8, перейдем в него и создадим файл lab8-1.asm: (рис. 2.1).

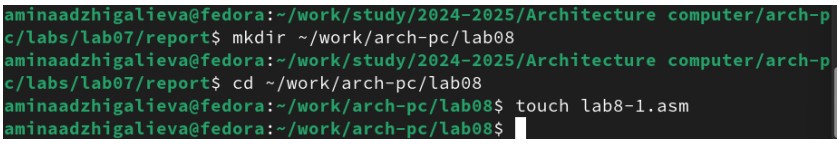

Рис. 2.1: Новый каталог

Введем в файл lab8-1.asm текст программы из листинга 8.1. (рис. 2.2).

```
%include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
mov eax,msg1
call sprint
mov ecx, N
mov edx, 10
call sread
mov eax,N
call atoi
mov [N],eax
mov ecx,[N]
label:
mov [N],ecx
mov eax,[N]
call iprintLF
loop label
call quit
```

Рис. 2.2: Текст программы

Создаем исполняемый файл и проверяем его работу. (рис. 2.3).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8
-1.o
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 8
8
7
6
5
4
3
2
1
```

Рис. 2.3: Проверка программы

Изменим текст программы добавив изменение значение регистра ecx в цикле:

(рис. 2.4).

```
%include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
mov eax,msg1
call sprint
mov ecx, N
mov edx, 10
call sread
mov eax,N
call atoi
mov [N],eax
mov ecx,[N]
label:
sub ecx, 1
mov [N],ecx
mov eax,[N]
call iprintLF
loop label
```

Рис. 2.4: Замена текста

Создаем исполняемый файл и проверяем его работу. (рис. 2.5).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8
-1.o
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 8
7
5
3
1
Ошибка сегментирования (образ памяти сброшен на диск)
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$
```

Рис. 2.5: Проверка программы

Регистр ecx принимает значения 7,5,3,1(на вход подается число 8, в цикле label данный регистр уменьшается на 2 командой sub и loop). Число проходов цикла не соответсвует числу N, так как уменьшается на 2

Внесем изменения в текст программы добавив команды push и pop (рис. 2.6).

```
%include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
mov eax,msg1
call sprint
mov ecx, N
mov edx, 10
call sread
mov eax,N
call atoi
mov [N],eax
mov ecx,[N]
label:
push ecx
sub ecx, 1
mov [N],ecx
mov eax,[N]
call iprintLF
pop ecx
loop label
```

Рис. 2.6: Заменим текст, чтобы все корректно работало

Создаем исполняемый файл и проверяем его работу. (рис. 2.7).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8
-1.o
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 8
7
6
5
4
3
2
1
0
Ошибка сегментирования (образ памяти сброшен на диск)
```

Рис. 2.7: Проверка программы

В данном случае число проходов цикла равна числу N.

2.2 Обработка аргументов командной строки

Создаем новый файл (рис. 2.8).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ touch lab8-2.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$
```

Рис. 2.8: Новый файл

Введем в него текст программы из листинга 8.2. (рис. 2.9).

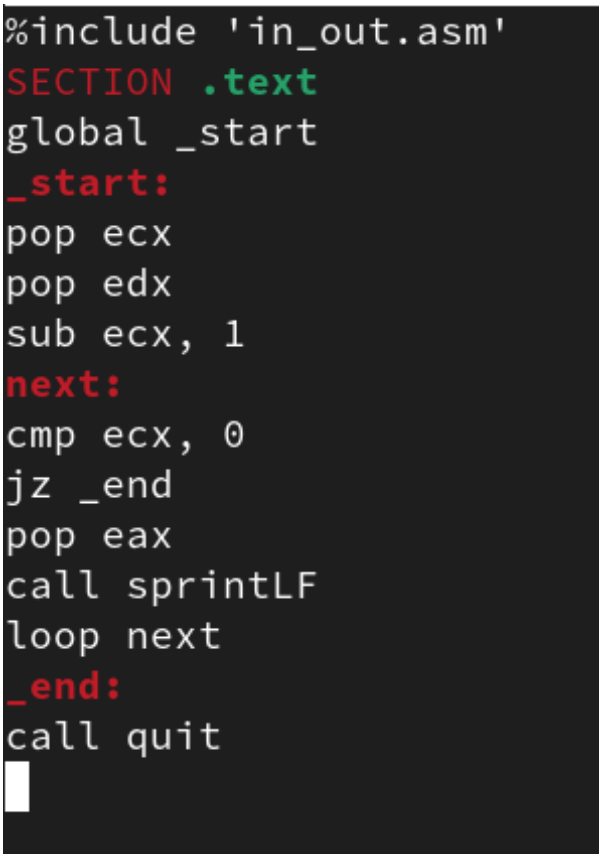

Рис. 2.9: Текст программы

Создаем исполняемый файл и проверяем его работу, указав аргументы. (рис. 2.10).

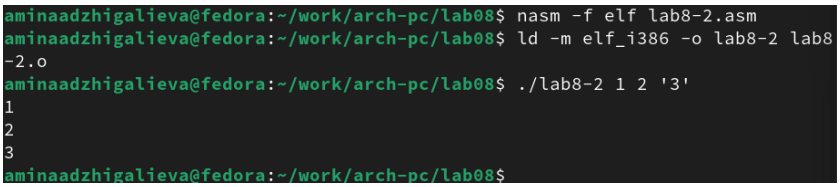

Рис. 2.10: Проверка программы

Програмой было обработано 3 аргумента.

Создаем файл lab8-3.asm в каталоге ~/work/archpc/lab08 (рис. 2.11).

Рис. 2.11: Новый файл

Введем в него текст программы из листинга 8.3. (рис. 2.12).

```
%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
pop ecx
pop edx
sub ecx,1
mov esi, 0
next:
cmp ecx,0h
jz _end
pop eax
call atoi
add esi,eax
loop next
_end:
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit
```

Рис. 2.12: Текст программы

Создаем исполняемый файл и проверяем его работу, указав аргументы. (рис. 2.13).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8
-3.o
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 5
Результат: 47
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$
```

Рис. 2.13: Проверка программы

Изменим текст программы из листинга 8.3 для вычисления произведения аргументов командной строки. (рис. 2.14).

```
%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
pop ecx
pop edx
sub ecx, 1
mov esi, 1
next:
cmp ecx, 0
jz _end
pop eax
call atoi
imul esi, eax
dec ecx
jmp next
_end:
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit
```

Рис. 2.14: Замена текста

Создаем исполняемый файл и проверяем его работу, указав аргументы. (рис. 2.15).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8
-3.o
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ./lab8-3 2 4 3
Результат: 24
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$
```

Рис. 2.15: Проверка программы

2.3 Задание для самостоятельной работы

Вариант 4. Функция: 2(x – 1)

Создаем файл (рис. 2.16).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ touch lab8-4.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$
```

Рис. 2.16: Новый файл

Пишем программу для нашей функции (рис. 2.17).

```
%include 'in_out.asm'
SECTION .data
msg db 'Сумма значений функции: ', 0
SECTION .text
global _start
_start:
pop ecx
pop edx
sub ecx, 1
mov esi, 0
next:
cmp ecx, 0
jz done
pop eax
call atoi
sub eax, 1
imul eax, eax, 2
add esi, eax
dec ecx
jmp next
done:
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit
```

Рис. 2.17: Текст программы

Создаем исполняемый файл и проверяем его работу, указав аргументы. (рис. 2.18).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8
-4.o
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ./lab8-4 3 5 7
Сумма значений функции: 24
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8
-4.o
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$ ./lab8-4 6 8 2
Сумма значений функции: 26
aminaadzhigalieva@fedora:~/work/arch-pc/lab08$
```

Рис. 2.18: Проверка программы

3 Выводы

Мы научились решать программы с использованием циклов и обработкой аргументов командной строки.