

Отчёт по лабораторной работе №8

Дисциплина: архитектура компьютера

Жернаков Данила Иванович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Реализация циклов в NASM	8
4.2	Обработка аргументов командной строки	10
4.3	Задание для самостоятельной работы (Вариант 5)	11
5	Выводы	13
	Список литературы	14

Список иллюстраций

4.1	Создание каталога и файла в нем	8
4.2	Исполнение программы	8
4.3	Изменение текста программы	9
4.4	Исполнение программы	9
4.5	Исполнение программы	9
4.6	Создание файла и его редактирование	10

Список таблиц

1 Цель работы

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

2 Задание

1. Реализация циклов в NASM
2. Обработка аргументов командной строки
3. Задание для самостоятельной работы

3 Теоретическое введение

Цикл в программировании — это управляющая конструкция, которая заставляет какой-то блок кода выполняться несколько раз.

4 Выполнение лабораторной работы

4.1 Реализация циклов в NASM

Создаю каталог для программ для лабораторной работы №8, перехожу в него и создаю файл lab8-1.asm, в который ввожу программу вывода значений регистра есх (рис. 4.1).

```
dizhernakov@dk2n26 ~ $ cd work/study/2023-2024/Архитектура\ компьютера/arch-pc/labsasm/  
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm $ mkdir lab08  
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm $ cd lab08  
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ touch lab8-1.asm  
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ ls  
lab8-1.asm  
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ cp ~/Загрузки/in_out.asm in_out.asm  
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ ls  
in_out.asm lab8-1.asm  
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $
```

Рис. 4.1: Создание каталога и файла в нем

Создаю исполняемый файл и запускаю его (рис. 4.2). Программа отработала корректно.

```
20 ; ----- Организация цикла  
21 mov ecx,[N] ; Счетчик цикла, 'ecx=N'  
22 label:  
23 sub ecx,1 ; 'ecx=ecx-1'  
24 mov [N],ecx  
25 mov eax,[N]  
26 call iprintLF  
27 loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
```

Рис. 4.2: Исполнение программы

Изменяю текст программы добавив изменение значения регистра есх в цикле (рис. 4.3).


```

20 ; ----- Организация цикла
21 mov ecx,[N] ; Счетчик цикла, 'ecx=N'
22 label:
23 sub ecx,1 ; 'ecx=ecx-1'
24 mov [N],ecx
25 mov eax,[N]
26 call iprintLF
27 loop label ; 'ecx=ecx-1' и если 'ecx' не '0'

```

Рис. 4.3: Изменение текста программы

Создаю исполняемый файл и запускаю его (рис. 4.4). Число проходов цикла равно $N/2$.

```

dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ nasm -f elf lab8-1.asm
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ ./lab8-1
Введите N: 10
9
7
5
3
1

```

Рис. 4.4: Исполнение программы

Вношу изменения в текст программы добавив команды push и pop для сохранения значения счётчика loop(рис. ??).

Создаю исполняемый файл и проверяю его работу (рис. 4.5). Число проходов соответствует значению N введённому с клавиатуры.

```

dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0

```

Рис. 4.5: Исполнение программы

4.2 Обработка аргументов командной строки

Создаю файл lab8-2.asm и ввожу в него программу обработки аргументов командной строки (рис. ??).

[Запуск исполняемого файла](image/6.png){#fig:007 width=70%}

Создаю файл lab8-3.asm и ввожу в него программу вычисления суммы аргументов командной строки (рис. ??). Создаю исполняемый файл и запускаю его, указав аргументы 12, 13, 7, 10, 5 (рис. ??). Программа отработала корректно.

```
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ touch lab8-3.asm
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ nasm -f elf lab8-3.asm
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ ./lab8-3 12 13 7 10 5
Результат: 47
```

Создаю файл lab8-3-1.asm и ввожу в него программу вычисления произведения аргументов командной строки (рис. 4.6). Создаю исполняемый файл и запускаю его, указав аргументы 12 10 2. Программа отработала корректно.

```
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ nasm -f elf lab8-3-1.asm
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ ld -m elf_i386 -o lab8-3-1 lab8-3-1.o
dizhernakov@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labsasm/lab08 $ ./lab8-3-1 12 10 2
Результат: 240
```

Рис. 4.6: Создание файла и его редактирование

Текст программы из файла lab8-3-1.asm:

```
%include "in_out.asm"

SECTION .data
msg db 'результат: '

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
    mul esi
    mov esi, eax

loop next

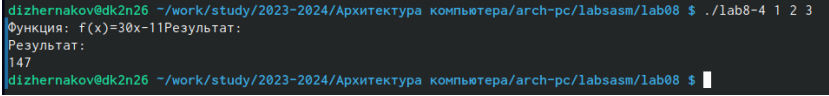
_end:
    mov eax, msg
    call sprint
    mov eax, esi
    call iprintLF
    call quit

```

4.3 Задание для самостоятельной работы (Вариант 5)

Создаю файл lab8-4.asm и ввожу текст программы для вычисления суммы значений функции $f(x)=30(30-x)$ (рис. ??). Создаю исполняемый файл и запускаю исполняемый файл с разными аргументами (рис. ??). Программа отработала

корректно.



Текст программы из файла lab8-4.asm:

```

#include 'in_out.asm'

SECTION .data
msg0 db "Функция: f(x)=30x-11"

```

```

msg1 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
mov edx, 30
mul edx
sub eax,11
add esi,eax
loop next
_end:
mov eax, msg0
call sprintfLF
mov eax, msg1
call sprintfLF
mov eax, esi
call sprintfLF
call quit

```

5 Выводы

Были получены навыки по организации циклов и работе со стеком на языке NASM.

Список литературы

1. Лабораторная работа №8