

Отчёт по лабораторной работе №8

Дисциплина: компьютерные науки и технологии программирования

Мирзоян Ян Игоревич

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Задание	7
4	Выполнение лабораторной работы	8
4.1	Реализация циклов в NASM	8
4.2	Обработка аргументов командной строки	11
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Создаю директорию, перехожу в нее и создаю необходимый файл, заполняю его	8
4.2	Создаю исполняемый файл и запускаю его, всё работает корректно	9
4.3	Изменяю файл, создаю исполняемый файл и запускаю его, всё работает корректно, число проходов цикла уменьшилось вдвое . . .	9
4.4	Изменяю файл, создаю исполняемый файл и запускаю его, всё работает корректно, число проходов цикла равно введенному с клавиатуры числу	10
4.5	Создаю необходимый файл и заполняю его	11
4.6	Создаю исполняемый файл и запускаю его, всё работает корректно	11
4.7	Создаю необходимый файл и заполняю его	12
4.8	Создаю исполняемый файл и запускаю его, всё работает корректно	12
4.9	Создаю необходимый файл и заполняю его	13
4.10	Создаю исполняемый файл и запускаю его, всё работает корректно	13

Список таблиц

1 Цель работы

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

2 Теоретическое введение

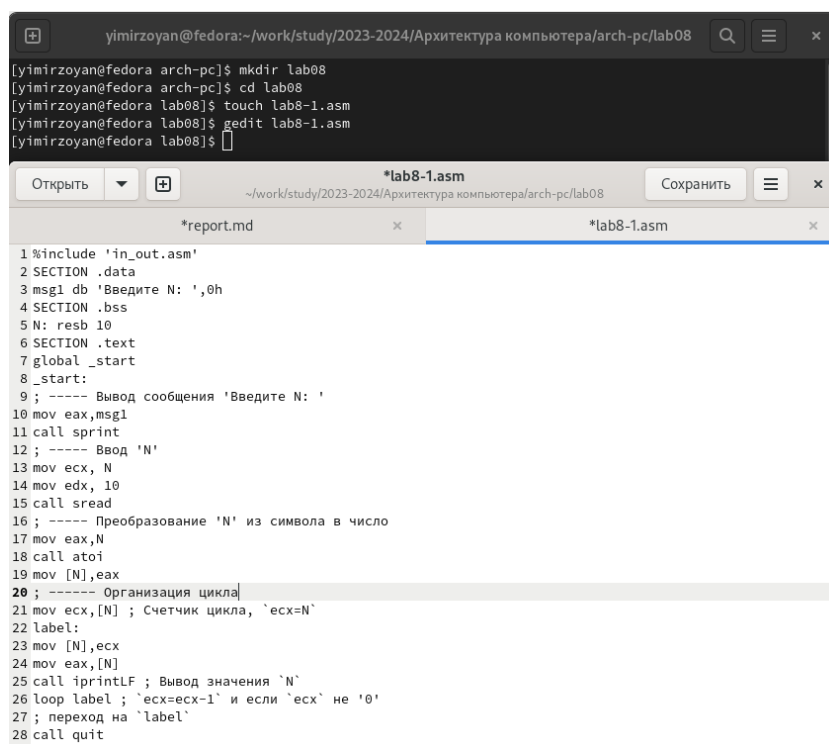
Стек — это структура данных, организованная по принципу LIFO («Last In — First Out» или «последним пришёл — первым ушёл»). Стек является частью архитектуры процессора и реализован на аппаратном уровне. Для работы со стеком в процессоре есть специальные регистры (ss, bp, sp) и команды. Основной функцией стека является функция сохранения адресов возврата и передачи аргументов при вызове процедур. Кроме того, в нём выделяется память для локальных переменных и могут временно храниться значения регистров.

3 Задание

1. Реализация циклов в NASM
2. Обработка аргументов командной строки

4 Выполнение лабораторной работы

4.1 Реализация циклов в NASM



The screenshot shows a terminal window at the top and a text editor window below it. The terminal window displays the following commands and their outputs:

```
[yimirzoyan@fedora arch-pc]$ mkdir lab08
[yimirzoyan@fedora arch-pc]$ cd lab08
[yimirzoyan@fedora lab08]$ touch lab8-1.asm
[yimirzoyan@fedora lab08]$ gedit lab8-1.asm
[yimirzoyan@fedora lab08]$
```

The text editor window shows the content of the file `lab8-1.asm`. The code is as follows:

```
1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Введите N: ',0h
4 SECTION .bss
5 N: resb 10
6 SECTION .text
7 global _start
8 _start:
9 ; ----- Вывод сообщения 'Введите N: '
10 mov eax,msg1
11 call sprint
12 ; ----- Ввод 'N'
13 mov ecx, N
14 mov edx, 10
15 call sread
16 ; ----- Преобразование 'N' из символа в число
17 mov eax,N
18 call atoi
19 mov [N],eax
20 ; ----- Организация цикла
21 mov ecx,[N] ; Счетчик цикла, 'ecx=N'
22 label:
23 mov [N],ecx
24 mov eax,[N]
25 call iprintfLF ; Вывод значения 'N'
26 loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
27 ; переход на 'label'
28 call quit
```

Рис. 4.1: Создаю директорию, перехожу в нее и создаю необходимый файл, заполняю его


```
[yimirzoyan@fedora lab08]$ nasm -f elf lab8-1.asm
[yimirzoyan@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[yimirzoyan@fedora lab08]$ ./lab8-1
Введите N: 10
10
9
8
7
6
5
4
3
2
1
[yimirzoyan@fedora lab08]$
```

Рис. 4.2: Создаю исполняемый файл и запускаю его, всё работает корректно

```
[yimirzoyan@fedora lab08]$ gedit lab8-1.asm
[yimirzoyan@fedora lab08]$ ^C
[yimirzoyan@fedora lab08]$ nasm -f elf lab8-1.asm
[yimirzoyan@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[yimirzoyan@fedora lab08]$ ./lab8-1
Введите N: 10
9
7
5
3
1
[yimirzoyan@fedora lab08]$
```

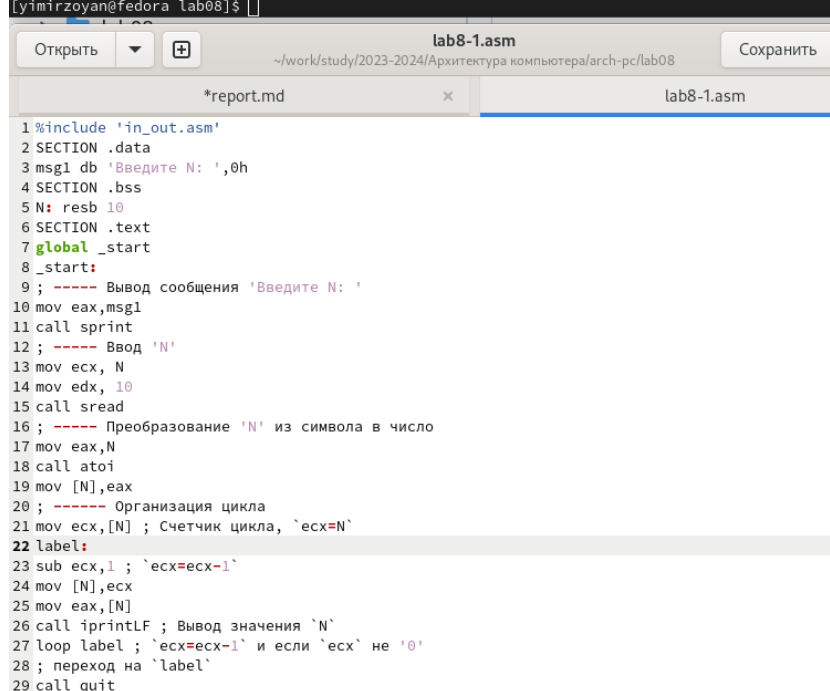


Рис. 4.3: Изменяю файл, создаю исполняемый файл и запускаю его, всё работает корректно, число проходов цикла уменьшилось вдвое

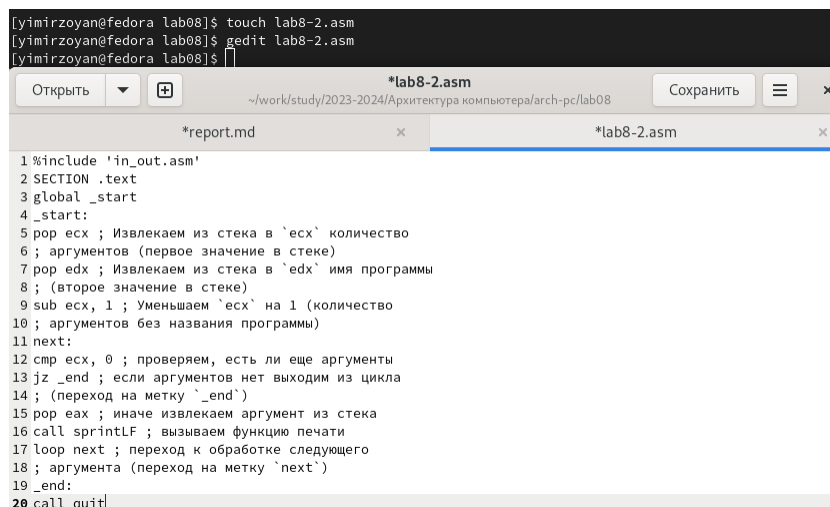
```

[yimirzoyan@fedora lab08]$ nasm -f elf lab8-1.asm
[yimirzoyan@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[yimirzoyan@fedora lab08]$ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
[yimirzoyan@fedora lab08]$
10 mov eax,msg1
11 call sprint
12 ; ----- Ввод 'N'
13 mov ecx, N
14 mov edx, 10
15 call sread
16 ; ----- Преобразование 'N' из символа в число
17 mov eax,N
18 call atoi
19 mov [N],eax
20 ; ----- Организация цикла
21 mov ecx,[N] ; Счетчик цикла, `ecx=N`
22 label:
23 push ecx ; добавление значения ecx в стек
24 sub ecx,1
25 mov [N],ecx
26 mov eax,[N]
27 call iprintLF
28 pop ecx ; извлечение значения ecx из стека
29 loop label
30 ; переход на `label`
31 call quit

```

Рис. 4.4: Изменяю файл, создаю исполняемый файл и запускаю его, всё работает корректно, число проходов цикла равно введенному с клавиатуры числу

4.2 Обработка аргументов командной строки



```
[yimirzoyan@fedora lab08]$ touch lab8-2.asm
[yimirzoyan@fedora lab08]$ gedit lab8-2.asm
[yimirzoyan@fedora lab08]$
```

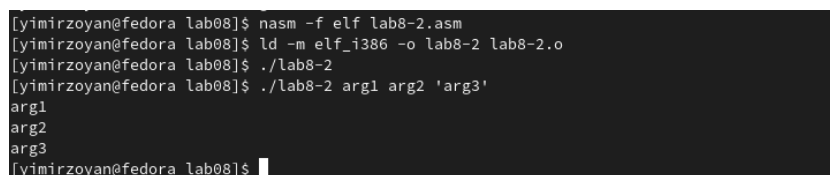
The screenshot shows a terminal window at the top with the following commands and their outputs:

```
[yimirzoyan@fedora lab08]$ touch lab8-2.asm
[yimirzoyan@fedora lab08]$ gedit lab8-2.asm
[yimirzoyan@fedora lab08]$
```

Below the terminal is a window of the gedit text editor. The title bar shows the file name `*lab8-2.asm` and the path `~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab08`. The editor contains the following assembly code:

```
1 %include 'in_out.asm'
2 SECTION .text
3 global _start
4 _start:
5 pop ecx ; Извлекаем из стека в 'ecx' количество
6 ; аргументов (первое значение в стеке)
7 pop edx ; Извлекаем из стека в 'edx' имя программы
8 ; (второе значение в стеке)
9 sub ecx, 1 ; Уменьшаем 'ecx' на 1 (количество
10 ; аргументов без названия программы)
11 next:
12 cmp ecx, 0 ; проверяем, есть ли еще аргументы
13 jz _end ; если аргументов нет выходим из цикла
14 ; (переход на метку '_end')
15 pop eax ; иначе извлекаем аргумент из стека
16 call sprintf ; вызываем функцию печати
17 loop next ; переход к обработке следующего
18 ; аргумента (переход на метку 'next')
19 _end:
20 call quit
```

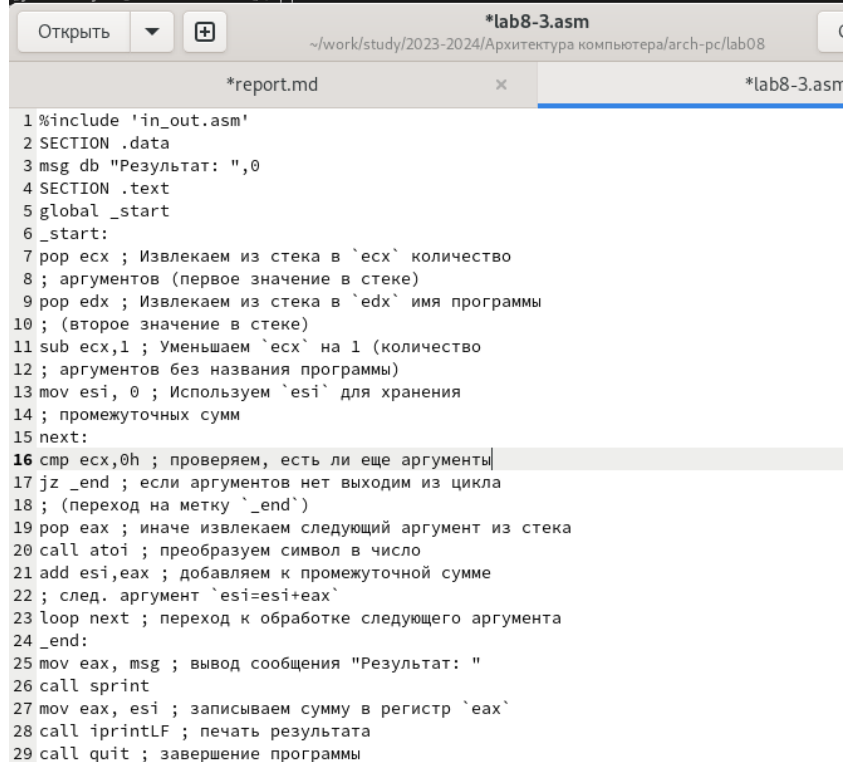
Рис. 4.5: Создаю необходимый файл и заполняю его



```
[yimirzoyan@fedora lab08]$ nasm -f elf lab8-2.asm
[yimirzoyan@fedora lab08]$ ld -m elf_i386 -o lab8-2 lab8-2.o
[yimirzoyan@fedora lab08]$ ./lab8-2
arg1
arg2
arg3
[yimirzoyan@fedora lab08]$
```

Рис. 4.6: Создаю исполняемый файл и запускаю его, всё работает корректно

```
[yimirzoyan@fedora lab08]$ touch lab8-3.asm
[yimirzoyan@fedora lab08]$ gedit lab8-3.asm
[yimirzoyan@fedora lab08]$
```



```
*lab8-3.asm
~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab08

*report.md x *lab8-3.asm

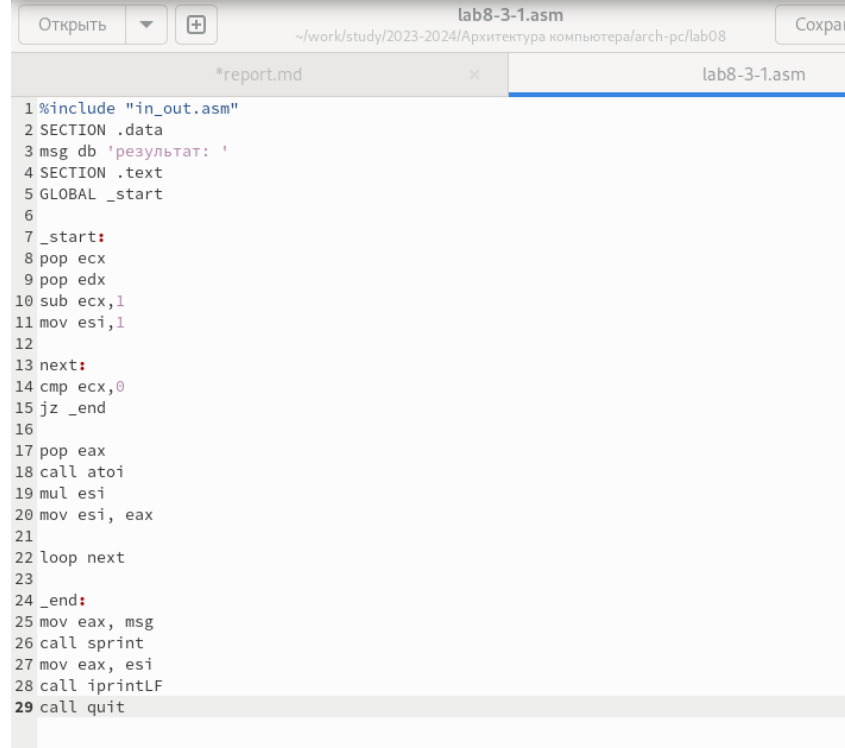
1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4 SECTION .text
5 global _start
6 _start:
7 pop ecx ; Извлекаем из стека в `ecx` количество
8 ; аргументов (первое значение в стеке)
9 pop edx ; Извлекаем из стека в `edx` имя программы
10 ; (второе значение в стеке)
11 sub ecx,1 ; Уменьшаем `ecx` на 1 (количество
12 ; аргументов без названия программы)
13 mov esi, 0 ; Используем `esi` для хранения
14 ; промежуточных сумм
15 next:
16 cmp ecx,0h ; проверяем, есть ли еще аргументы
17 jz _end ; если аргументов нет выходим из цикла
18 ; (переход на метку `_end`)
19 pop eax ; иначе извлекаем следующий аргумент из стека
20 call atoi ; преобразуем символ в число
21 add esi,eax ; добавляем к промежуточной сумме
22 ; след. аргумент `esi=esi+eax`
23 loop next ; переход к обработке следующего аргумента
24 _end:
25 mov eax, msg ; вывод сообщения "Результат: "
26 call sprint
27 mov eax, esi ; записываем сумму в регистр `eax`
28 call iprintLF ; печать результата
29 call quit ; завершение программы
```

Рис. 4.7: Создаю необходимый файл и заполняю его

```
[yimirzoyan@fedora lab08]$ nasm -f elf lab8-3.asm
[yimirzoyan@fedora lab08]$ ld -m elf_i386 -o lab8-3 lab8-3.o
[yimirzoyan@fedora lab08]$ ./lab8-3 12 13 7 10 5
Результат: 47
[yimirzoyan@fedora lab08]$
```

Рис. 4.8: Создаю исполняемый файл и запускаю его, всё работает корректно

```
[yimirzoyan@fedora lab08]$ touch lab8-3-1.asm
[yimirzoyan@fedora lab08]$ gedit lab8-3-1.asm
[yimirzoyan@fedora lab08]$ gedit lab8-3-1.asm
[yimirzoyan@fedora lab08]$
```



```
1 %include "in_out.asm"
2 SECTION .data
3 msg db 'результат: '
4 SECTION .text
5 GLOBAL _start
6
7 _start:
8 pop ecx
9 pop edx
10 sub ecx,1
11 mov esi,1
12
13 next:
14 cmp ecx,0
15 jz _end
16
17 pop eax
18 call atoi
19 mul esi
20 mov esi, eax
21
22 loop next
23
24 _end:
25 mov eax, msg
26 call sprint
27 mov eax, esi
28 call iprintLF
29 call quit
```

Рис. 4.9: Создаю необходимый файл и заполняю его

```
[yimirzoyan@fedora lab08]$ nasm -f elf lab8-3-1.asm
[yimirzoyan@fedora lab08]$ ld -m elf_i386 -o lab8-3-1 lab8-3-1.o
[yimirzoyan@fedora lab08]$ ./lab8-3-1 1 2 3 4 5
результат: 120
[yimirzoyan@fedora lab08]$
```

Рис. 4.10: Создаю исполняемый файл и запускаю его, всё работает корректно

5 Выводы

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

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс,
- 11.
12. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
13. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
14. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
15. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-

- е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.
16. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
 17. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).