

Отчет по лабораторной работе №7

Дисциплина: архитектура компьютера

Байдина Елизавета Дмитриевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Реализация переходов в NASM	8
4.2	Выполнение заданий для самостоятельной работы	13
5	Выводы	16
	Список литературы	17

Список иллюстраций

4.1	Создание каталога, переход в него и создание файла	8
4.2	Копирование файла	8
4.3	Текст программы листинга 7.1	9
4.4	Создание и запуск файла	9
4.5	Текст программы листинга 7.2	9
4.6	Создание исполняемого файла и запуск его	10
4.7	Изменение текста программы	10
4.8	Создание файла и запуск	10
4.9	Создание файла	10
4.10	Текст программы листинга 7.3	11
4.11	Проверка программы	11
4.12	Создание файла листинга	12
4.13	Объяснение первой строки	12
4.14	Объяснение второй строки	12
4.15	Объяснение третьей строки	12
4.16	Удаление одного из операндов	12
4.17	Трансляция с получением файла листинга	12
4.18	Создание файла	13
4.19	Текст программы задания №1	13
4.20	Создание файла и запуск	14
4.21	Создание файла	14
4.22	Текст программы задания №2	15
4.23	Создание файла и запуск	15

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . . .	7
-----	---	---

1 Цель работы

Целью данной лабораторной работы является изучение команд условного и безусловного переходов, приобретение навыков написания программ с использованием переходов, а также знакомство с назначением и структурой файла листинга.

2 Задание

1. Реализация переходов в NASM
2. Изучение структуры файлы листинга
3. Ответы на вопросы
4. Выполнение заданий для самостоятельной работы

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [1–4].

4 Выполнение лабораторной работы

4.1 Реализация переходов в NASM

Создаю каталог для программ лабораторной работы №7, перехожу в него и создаю файл lab7-1.asm (рис. 4.1)

```
edbayjdina@dk8n61 ~ $ mkdir ~/work/arch-pc/lab07
edbayjdina@dk8n61 ~ $ cd ~/work/arch-pc/lab07
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ touch lab7-1.asm
```

Рис. 4.1: Создание каталога, переход в него и создание файла

Копирую файл in_out.asm из загрузок в соответствующую папку для дальнейшей работы (рис. 4.2)

```
edbayjdina@dk3n35 ~ $ cd ~/work/arch-pc/lab07
edbayjdina@dk3n35 ~/work/arch-pc/lab07 $ ls
in_out.asm  lab7-1.o    lab7-2.lst  lab7-3.o    lab7-4.o
lab7-1      lab7-2      lab7-3      lab7-4
lab7-1.asm  lab7-2.asm  lab7-3.asm  lab7-4.asm
edbayjdina@dk3n35 ~/work/arch-pc/lab07 $ █
```

Рис. 4.2: Копирование файла

Ввожу в lab7-1.asm текст программы из листинга 7.1 (рис. 4.3)


```

1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 msg1: DB 'Сообщение № 1',0
4 msg2: DB 'Сообщение № 2',0
5 msg3: DB 'Сообщение № 3',0
6 SECTION .text
7 GLOBAL _start
8 _start:
9 jmp _label2
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintf ; 'Сообщение № 1'
13 _label2:
14 mov eax, msg2 ; Вывод на экран строки
15 call sprintf ; 'Сообщение № 2'
16 _label3:
17 mov eax, msg3 ; Вывод на экран строки
18 call sprintf ; 'Сообщение № 3'
19 _end:
20 call quit ; вызов подпрограммы завершения

```

Рис. 4.3: Текст программы листинга 7.1

Создаю исполняемый файл и запускаю его (рис. 4.4)

```

edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 3

```

Рис. 4.4: Создание и запуск файла

Ввожу текст программы листинга 7.2 (рис. 4.5)

```

1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 msg1: DB 'Сообщение № 1',0
4 msg2: DB 'Сообщение № 2',0
5 msg3: DB 'Сообщение № 3',0
6 SECTION .text
7 GLOBAL _start
8 _start:
9 jmp _label2
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintf ; 'Сообщение № 1'
13 jmp _end
14 _label2:
15 mov eax, msg2 ; Вывод на экран строки
16 call sprintf ; 'Сообщение № 2'
17 jmp _label1
18 _label3:
19 mov eax, msg3 ; Вывод на экран строки
20 call sprintf ; 'Сообщение № 3'
21 _end:
22 call quit ; вызов подпрограммы завершения

```

Рис. 4.5: Текст программы листинга 7.2

Создаю исполняемый файл и запускаю его (рис. 4.6)

```
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 1
```

Рис. 4.6: Создание исполняемого файла и запуск его

Изменяю текст программы так, чтоб вывод был в нужной последовательности (рис. 4.7)

```
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 msg1: DB 'Сообщение № 1',0
4 msg2: DB 'Сообщение № 2',0
5 msg3: DB 'Сообщение № 3',0
6 SECTION .text
7 GLOBAL _start
8 _start:
9 jmp _label3
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintf ; 'Сообщение № 1'
13 jmp _end
14 _label2:
15 mov eax, msg2 ; Вывод на экран строки
16 call sprintf ; 'Сообщение № 2'
17 jmp _label1
18 _label3:
19 mov eax, msg3 ; Вывод на экран строки
20 call sprintf ; 'Сообщение № 3'
21 jmp _label2
22 _end:
23 call quit ; вызов подпрограммы завершения
```

Рис. 4.7: Изменение текста программы

Создаю исполняемый файл и запускаю его (рис. 4.8)

```
edbayjdina@dk3n35 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
edbayjdina@dk3n35 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
edbayjdina@dk3n35 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
```

Рис. 4.8: Создание файла и запуск

Создаю файл lab7-2.asm (рис. 4.9)

```
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ touch lab7-2.asm
```

Рис. 4.9: Создание файла

Ввожу текст программы из листинга 7.3 в файл (рис. 4.10)

```
lab7-1.asm
1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите B: ',0h
4 msg2 db "Наибольшее число: ",0h
5 A dd '20'
6 C dd '50'
7 section .bss
8 max resb 10
9 B resb 10
10 section .text
11 global _start
12 _start:
13 ; ----- Вывод сообщения 'Введите B: '
14 mov eax,msg1
15 call sprint
16 ; ----- Ввод 'B'
17 mov ecx,B
18 mov edx,10
19 call sread
20 ; ----- Преобразование 'B' из символа в число
21 mov eax,B
22 call atoi ; Вызов подпрограммы перевода символа в число
23 mov [B],eax ; запись преобразованного числа в 'B'
24 ; ----- Записываем 'A' в переменную 'max'
25 mov ecx,[A] ; 'ecx = A'
26 mov [max],ecx ; 'max = A'
27 ; ----- Сравниваем 'A' и 'C' (как символы)
28 cmp ecx,[C] ; Сравниваем 'A' и 'C'
29 jg check_B ; если 'A>C', то переход на метку 'check_B',
30 mov ecx,[C] ; иначе 'ecx = C'
31 mov [max],ecx ; 'max = C'
32 ; ----- Преобразование 'max(A,C)' из символа в число
33 check_B:
34 mov eax,max
35 call atoi ; Вызов подпрограммы перевода символа в число
36 mov [max],eax ; запись преобразованного числа в 'max'
37 ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
38 mov ecx,[max]
39 cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'
40 jg fin ; если 'max(A,C)>B', то переход на 'fin',
41 mov ecx,[B] ; иначе 'ecx = B'
42 mov [max],ecx
43 ; ----- Вывод результата
44 fin:
45 mov eax, msg2
```

Рис. 4.10: Текст программы листинга 7.3

Создаю исполняемый файл и проверяю его работу для разных значений B (рис. 4.11)

```
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ ./lab7-2
Введите B: 4
Наибольшее число: 50
```

Рис. 4.11: Проверка программы

##Изучение структуры файлы листинга

Создаю файл листинга для программы из файла lab7-2.asm и открываю его (рис. 4.12)

```
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ nasm -f elf -l lab7-2.lst lab7-2.asm
```

Рис. 4.12: Создание файла листинга

Объяснение строки: В регистр `eax` мы вносим значение 4 (рис. 4.13)

```
34 00000022 B804000000 <1> mov eax, 4
```

Рис. 4.13: Объяснение первой строки

Объяснение строки: В регистр `eax` мы вносим значение 3 (рис. 4.14)

```
66 0000004A B803000000 <1> mov eax, 3
```

Рис. 4.14: Объяснение второй строки

Объяснение строки: Вызов подпрограммы перевода символа в число (рис. 4.15)

```
35 00000135 E862FFFFFF call atoi ; Вызов подпрограммы перевода символа в число
```

Рис. 4.15: Объяснение третьей строки

Удаляю один из операндов (рис. 4.16)

	lab7-1.asm	×	lab7-2.asm
181	6 00000039 35300000		C dd '50'
182	7		section .bss
183	8 00000000 <res Ah>		max resb 10
184	9 0000000A <res Ah>		B resb 10
185	10		section .text
186	11		global _start
187	12		_start:
188	13		; ----- Вывод сообщения 'Введите B: '
189	14		mov eax
190	14 *****		error: invalid combination of opcode and operands

Рис. 4.16: Удаление одного из операндов

Выполняю трансляцию с получением файла листинга, но ничего не выводит (рис. 4.17)

```
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:14: error: invalid combination of opcode and operands
```

Рис. 4.17: Трансляция с получением файла листинга

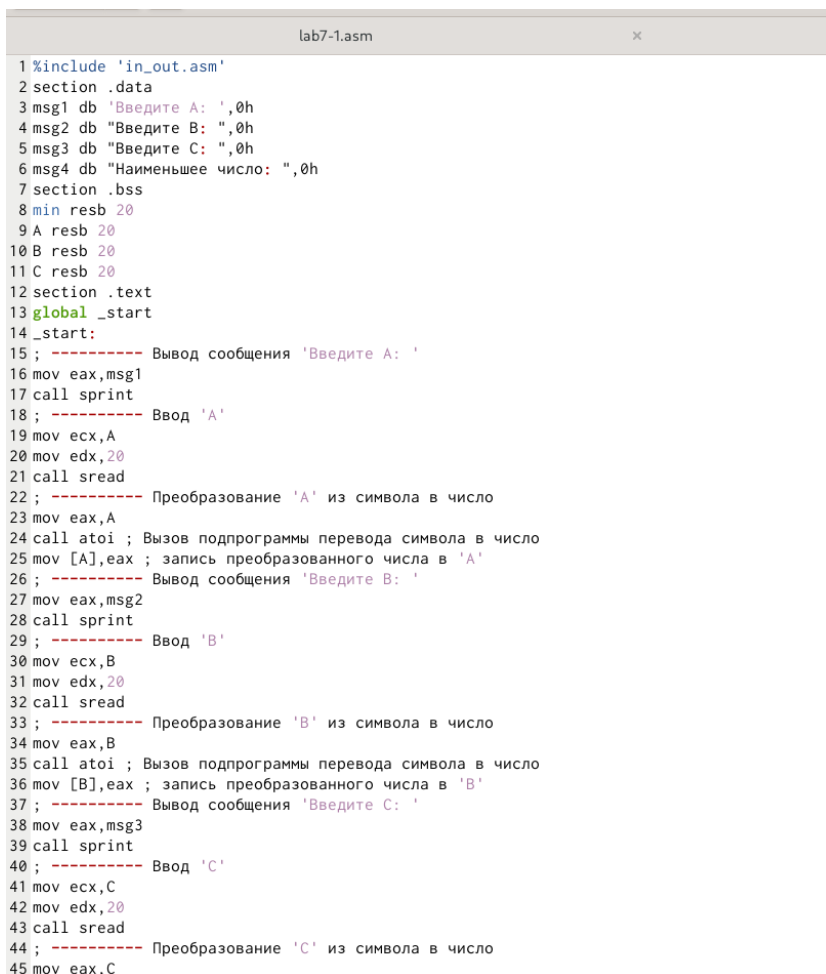
4.2 Выполнение заданий для самостоятельной работы

Создаю файл для задания №1 (рис. 4.18)

```
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ touch lab7-3.asm
```

Рис. 4.18: Создание файла

Записываю текст программы для нахождения наименьшего числа (рис. 4.19)



```
lab7-1.asm
1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите A: ',0h
4 msg2 db "Введите B: ",0h
5 msg3 db "Введите C: ",0h
6 msg4 db "Наименьшее число: ",0h
7 section .bss
8 min resb 20
9 A resb 20
10 B resb 20
11 C resb 20
12 section .text
13 global _start
14 _start:
15 ; ----- Вывод сообщения 'Введите A: '
16 mov eax,msg1
17 call sprint
18 ; ----- Ввод 'A'
19 mov ecx,A
20 mov edx,20
21 call sread
22 ; ----- Преобразование 'A' из символа в число
23 mov eax,A
24 call atoi ; Вызов подпрограммы перевода символа в число
25 mov [A],eax ; запись преобразованного числа в 'A'
26 ; ----- Вывод сообщения 'Введите B: '
27 mov eax,msg2
28 call sprint
29 ; ----- Ввод 'B'
30 mov ecx,B
31 mov edx,20
32 call sread
33 ; ----- Преобразование 'B' из символа в число
34 mov eax,B
35 call atoi ; Вызов подпрограммы перевода символа в число
36 mov [B],eax ; запись преобразованного числа в 'B'
37 ; ----- Вывод сообщения 'Введите C: '
38 mov eax,msg3
39 call sprint
40 ; ----- Ввод 'C'
41 mov ecx,C
42 mov edx,20
43 call sread
44 ; ----- Преобразование 'C' из символа в число
45 mov eax,C
```

Рис. 4.19: Текст программы задания №1

Создаю исполняемый файл и проверяю правильность работы (рис. 4.20)

```
edbayjdina@dk3n35 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
edbayjdina@dk3n35 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
edbayjdina@dk3n35 ~/work/arch-pc/lab07 $ ./lab7-3
Введите A: 84
Введите B: 32
Введите C: 77
Наименьшее число: 32
```

Рис. 4.20: Создание файла и запуск

Создаю файл для задания №2 (рис. 4.21)

```
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ touch lab7-4.asm
```

Рис. 4.21: Создание файла

Записываю текст программы для вычисления выражения $f(x)$ (рис. 4.22)

```

1 %include 'in_out.asm' ; подключение внешнего файла
2 section .data
3     msg1 db 'Введите A: ',0h
4     msg2 db 'Введите X: ',0h
5     msg3 db "Результат: ",0h
6
7 section .bss
8     fin resb 10
9     A resb 10
10    X resb 10
11
12 section .text
13     global _start
14
15 _start:
16 ; ----- Вывод сообщения 'Введите A: '
17     mov eax,msg1
18     call sprint
19 ; ----- Ввод 'A'
20     mov ecx,A
21     mov edx,10
22     call sread
23 ; ----- Преобразование 'A' из символа в число
24     mov eax,A
25     call atoi ; Вызов подпрограммы перевода символа в число
26     mov [A],eax
27
28 ; ----- Вывод сообщения 'Введите X: '
29     mov eax,msg2
30     call sprint
31 ; ----- Ввод 'X'
32     mov ecx,X
33     mov edx,10
34     call sread
35 ; ----- Преобразование 'X' из символа в число
36     mov eax,X
37     call atoi ; Вызов подпрограммы перевода символа в число
38     mov [X],eax
39
40
41     mov ecx,[A]
42     cmp ecx,7
43     jge func2
44     mov edx,[X]
45     mov ax,[A]

```

Сохранение файла «~/work/arch-pc/lab07/lab7-4.asm»...

Рис. 4.22: Текст программы задания №2

Создаю исполняемый файл, запускаю и убеждаюсь в правильности программы (рис. 4.23)

```

edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ touch lab7-4.asm
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ nasm -f elf lab7-4.asm
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-4 lab7-4.o
edbayjdina@dk8n61 ~/work/arch-pc/lab07 $ ./lab7-4
Введите A: 9
Введите X: 3
Результат: 2

```

Рис. 4.23: Создание файла и запуск

5 Выводы

Я ознакомилась с командами условного и безусловного переходов, приобрела навыки написания программ с использованием переходов, а также познакомилась с назначением и структурой файла листинга.

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.