

Отчёт по лабораторной работе

Дисциплина: Архитектура ЭВМ

Перегудов Александр Вадимович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	21
	Список литературы	22

Список иллюстраций

4.1	Создание каталога с помощью mkdir и файла с помощью touch . .	8
4.2	Открытие файла с gedit	8
4.3	Текст программы	9
4.4	Результат работы программы	9
4.5	Изменённый текст программы	10
4.6	Результат работы программы.....	10
4.7	Изменённый текст программы	11
4.8	Результат работы программы.....	11
4.9	Создание файла с помощью touch	12
4.10	Текст программы.....	12
4.11	Трансляция и компоновка.....	12
4.12	Результат работы программы.....	13
4.13	Результат работы программы.....	13
4.14	Создание файла листинга с помощью ключа -l	13
4.15	Открытие файла с gedit	13
4.16	Текст файла листинга lab7-2.asm	14
4.17	Изменённый текст программы.....	15
4.18	Результат трансляции.....	15
4.19	Текст файла листинга lab7-2.asm	16
4.20	Создал файл lab7-3.asm.....	16
4.21	Текст программы.....	17
4.22	Продолжение текста.....	18
4.23	Результат работы программы.....	18
4.24	Создал файл lab7-3.asm.....	18
4.25	Текст программы.....	19
4.26	Трансляция и компоновка.....	19
4.27	Результат работы программы.....	20
4.28	Результат работы программы.....	20

Список таблиц

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

1 Цель работы

Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

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 <code>root</code>
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно об Unix см. в [1–6].

4 Выполнение лабораторной работы

1. Создал каталог для программ лабораторной работы No 7, перешёл в него и создал файл lab7-1.asm (рис. 4.1).

```
avperegudov@dk8n76 ~ $ mkdir ~/work/arch-pc/lab07
avperegudov@dk8n76 ~ $ cd ~/work/arch-pc/lab07
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ touch lab7-1.asm
avperegudov@dk8n76 ~/work/arch-pc/lab07 $
```

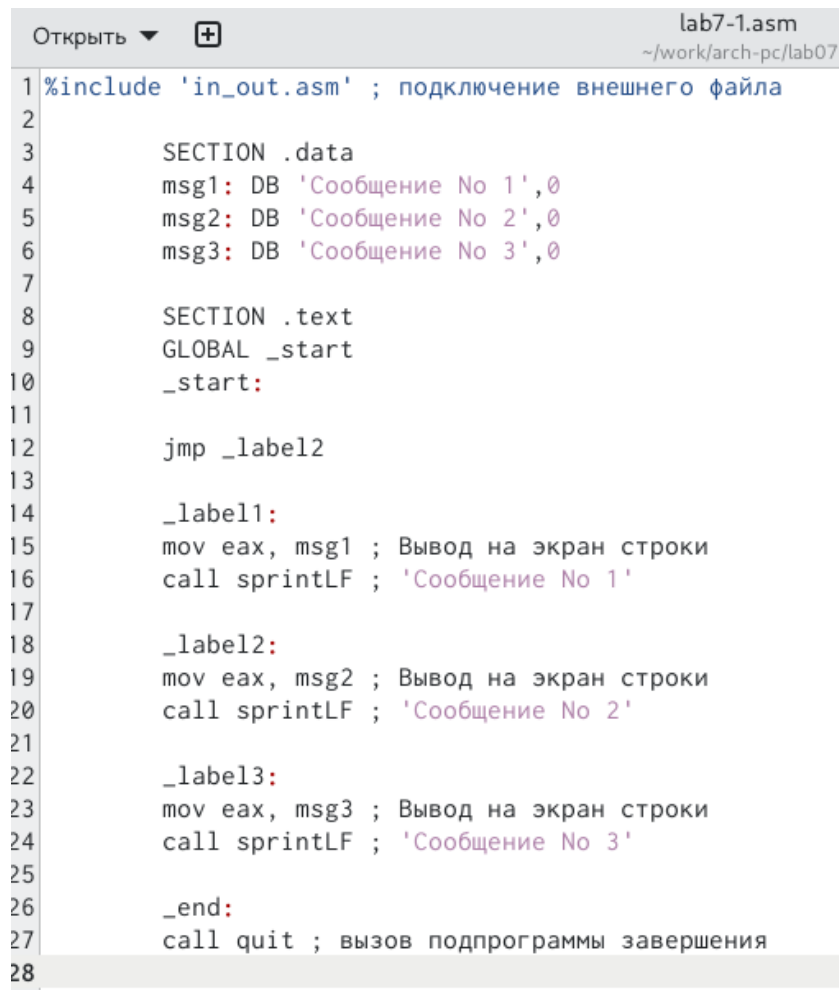
Рис. 4.1: Создание каталога с помощью mkdir и файла с помощью touch

2. Открыл файл lab7-1.asm с помощью gedit (рис. 4.2).

```
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ gedit lab7-1.asm
```

Рис. 4.2: Открытие файла с gedit

3. Ввёл в файл lab7-1.asm текст программы из листинга 7.1 (рис. 4.3).



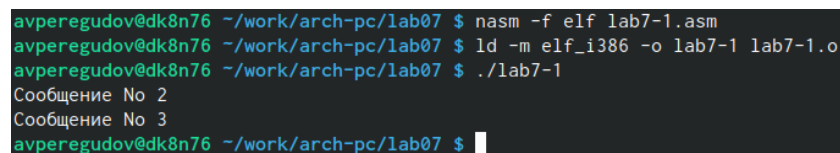
```

1 %include 'in_out.asm' ; подключение внешнего файла
2
3     SECTION .data
4     msg1: DB 'Сообщение No 1',0
5     msg2: DB 'Сообщение No 2',0
6     msg3: DB 'Сообщение No 3',0
7
8     SECTION .text
9     GLOBAL _start
10    _start:
11
12    jmp _label2
13
14    _label1:
15    mov eax, msg1 ; Вывод на экран строки
16    call sprintf ; 'Сообщение No 1'
17
18    _label2:
19    mov eax, msg2 ; Вывод на экран строки
20    call sprintf ; 'Сообщение No 2'
21
22    _label3:
23    mov eax, msg3 ; Вывод на экран строки
24    call sprintf ; 'Сообщение No 3'
25
26    _end:
27    call quit ; вызов подпрограммы завершения
28

```

Рис. 4.3: Текст программы

4. Создал исполняемый файл и запустил его (рис. 4.4).



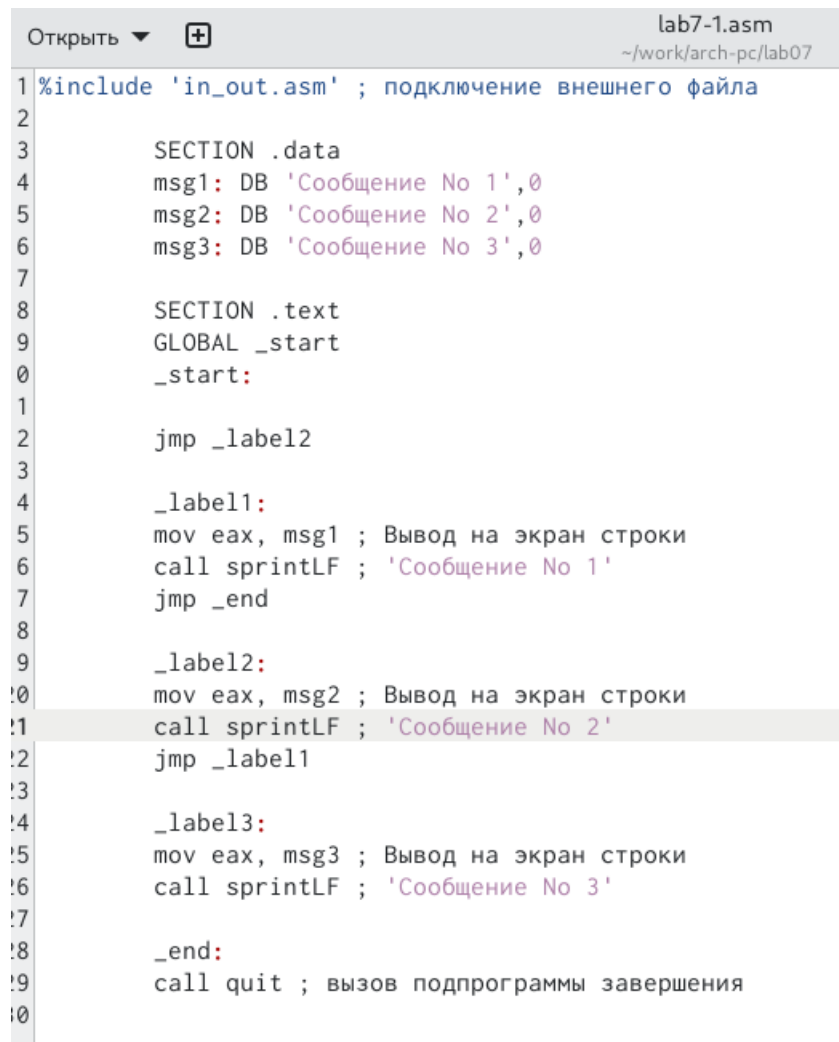
```

avperegudov@dk8n76 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение No 2
Сообщение No 3
avperegudov@dk8n76 ~/work/arch-pc/lab07 $

```

Рис. 4.4: Результат работы программы

5. Изменил текст программы в соответствии с листингом 7.2 (рис. 4.5).

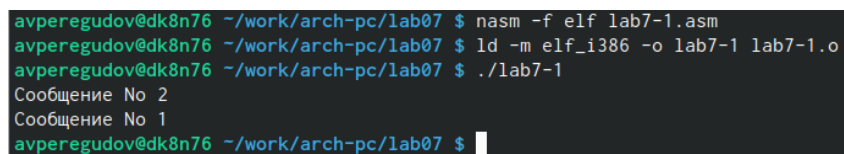


```
lab7-1.asm
~/work/arch-pc/lab07

1 %include 'in_out.asm' ; подключение внешнего файла
2
3     SECTION .data
4     msg1: DB 'Сообщение No 1',0
5     msg2: DB 'Сообщение No 2',0
6     msg3: DB 'Сообщение No 3',0
7
8     SECTION .text
9     GLOBAL _start
10    _start:
11
12    jmp _label2
13
14    _label1:
15    mov eax, msg1 ; Вывод на экран строки
16    call sprintf ; 'Сообщение No 1'
17    jmp _end
18
19    _label2:
20    mov eax, msg2 ; Вывод на экран строки
21    call sprintf ; 'Сообщение No 2'
22    jmp _label1
23
24    _label3:
25    mov eax, msg3 ; Вывод на экран строки
26    call sprintf ; 'Сообщение No 3'
27
28    _end:
29    call quit ; вызов подпрограммы завершения
30
```

Рис. 4.5: Изменённый текст программы

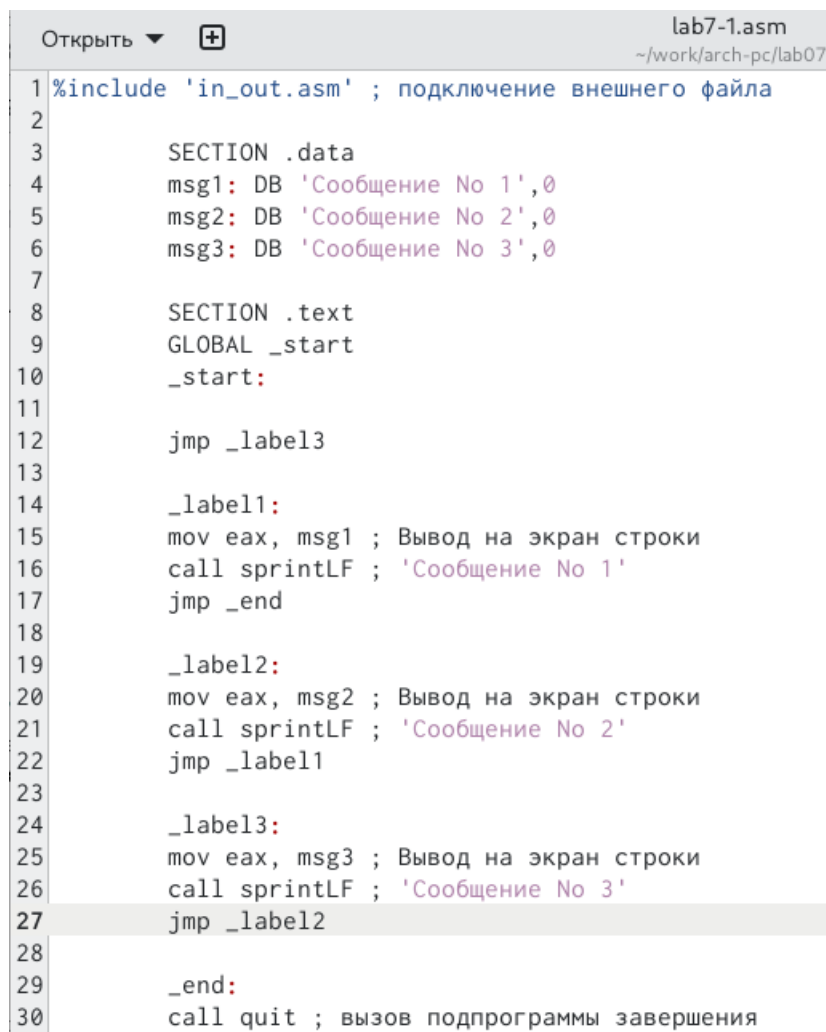
6. Создал исполняемый файл и запустил его (рис. 4.6).



```
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение No 2
Сообщение No 1
avperegudov@dk8n76 ~/work/arch-pc/lab07 $
```

Рис. 4.6: Результат работы программы

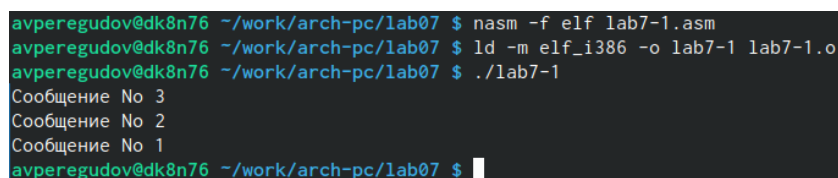
7. Изменил текст программы в соответствии с требованием вывести числа 3, 2, 1 (рис. 4.7).



```
Открыть + lab7-1.asm
~/work/arch-pc/lab07
1 %include 'in_out.asm' ; подключение внешнего файла
2
3     SECTION .data
4     msg1: DB 'Сообщение No 1',0
5     msg2: DB 'Сообщение No 2',0
6     msg3: DB 'Сообщение No 3',0
7
8     SECTION .text
9     GLOBAL _start
10    _start:
11
12    jmp _label3
13
14    _label1:
15    mov eax, msg1 ; Вывод на экран строки
16    call sprintf ; 'Сообщение No 1'
17    jmp _end
18
19    _label2:
20    mov eax, msg2 ; Вывод на экран строки
21    call sprintf ; 'Сообщение No 2'
22    jmp _label1
23
24    _label3:
25    mov eax, msg3 ; Вывод на экран строки
26    call sprintf ; 'Сообщение No 3'
27    jmp _label2
28
29    _end:
30    call quit ; вызов подпрограммы завершения
```

Рис. 4.7: Изменённый текст программы

8. Создал исполняемый файл и запустил его (рис. 4.8).



```
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение No 3
Сообщение No 2
Сообщение No 1
avperegudov@dk8n76 ~/work/arch-pc/lab07 $
```

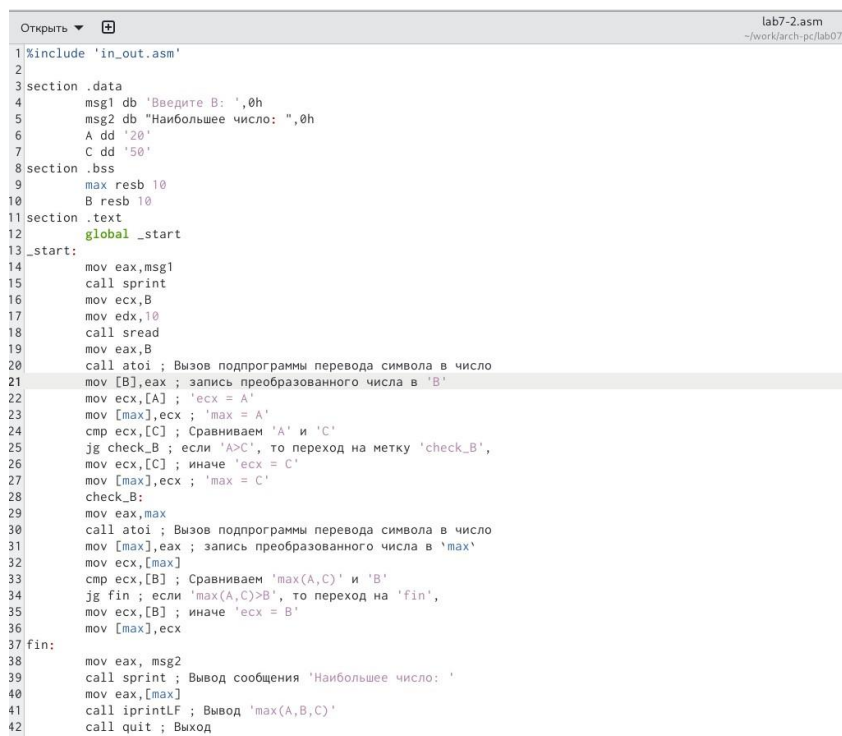
Рис. 4.8: Результат работы программы

9. Создал файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 (рис. 4.9).

```
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ touch lab7-2.asm
avperegudov@dk8n76 ~/work/arch-pc/lab07 $
```

Рис. 4.9: Создание файла с помощью touch

10. Ввёл в файл lab7-2.asm текст программы из листинга 7.3 (рис. 4.10).



```

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

```

Рис. 4.10: Текст программы

11. Создал исполняемый файл (рис. 4.11).

```
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
```

Рис. 4.11: Трансляция и компоновка

12. Запустил исполняемый файл со значением B=4 (рис. 4.12).

```
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ ./lab7-2
Введите В: 4
Наибольшее число: 50
```

Рис. 4.12: Результат работы программы

13. Запустил исполняемый файл со значением В=99 (рис. 4.13).

```
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ ./lab7-2
Введите В: 99
Наибольшее число: 99
```

Рис. 4.13: Результат работы программы

14. Создал файл листинга для программы из файла lab7-2.asm (рис. 4.14).

```
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ nasm -f elf -l lab7-2.lst lab7-2.asm
avperegudov@dk8n76 ~/work/arch-pc/lab07 $
```

Рис. 4.14: Создание файла листинга с помощью ключа -l

15. Открыл файл листинга lab7-2.lst с помощью gedit (рис. 4.15, 4.16).

```
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ gedit lab7-2.lst
avperegudov@dk8n76 ~/work/arch-pc/lab07 $
```

Рис. 4.15: Открытие файла с gedit

```

Открыть  lab7-2.lst
~\work\arch-pc\lab07

1 1 %include 'in_out.asm'
2 2 <!------- slen ----->
3 3 <!-- Функция вычисления длины сообщения -->
4 4 <!-- slen: -->
5 5 00000000 53 <!-- push ebx -->
6 6 00000001 89C3 <!-- mov ebx, eax -->
7 7 <!-- -->
8 8 <!-- nextchar: -->
9 9 00000003 803800 <!-- cmp byte [eax], 0 -->
10 10 00000006 7403 <!-- jz finished -->
11 11 00000008 40 <!-- inc eax -->
12 12 00000009 EBF8 <!-- jmp nextchar -->
13 13 <!-- -->
14 14 <!-- finished: -->
15 15 0000000B 29D8 <!-- sub eax, ebx -->
16 16 0000000D 5B <!-- pop ebx -->
17 17 0000000E C3 <!-- ret -->
18 18 <!-- -->
19 19 <!-- -->
20 20 <!-- ;----- sprint ----->
21 21 <!-- Функция печати сообщения -->
22 22 <!-- входные данные: mov eax,<message> -->
23 23 <!-- sprint: -->
24 24 0000000F 52 <!-- push edx -->
25 25 00000010 51 <!-- push ecx -->
26 26 00000011 53 <!-- push ebx -->
27 27 00000012 50 <!-- push eax -->
28 28 00000013 E8E8FFFFFF <!-- call slen -->
29 29 <!-- -->
30 30 00000018 89C2 <!-- mov edx, eax -->
31 31 0000001A 58 <!-- pop eax -->
32 32 <!-- -->
33 33 0000001B 89C1 <!-- mov ecx, eax -->
34 34 0000001D BB01000000 <!-- mov ebx, 1 -->
35 35 00000022 B804000000 <!-- mov eax, 4 -->
36 36 00000027 CD80 <!-- int 80h -->
37 37 <!-- -->
38 38 00000029 5B <!-- pop ebx -->
39 39 0000002A 59 <!-- pop ecx -->
40 40 0000002B 5A <!-- pop edx -->
41 41 0000002C C3 <!-- ret -->
42 42 <!-- -->
43 43 <!-- -->
44 44 <!-- ;----- sprintf ----->
45 45 <!-- Функция печати сообщения с переводом строки -->
46 46 <!-- входные данные: mov eax,<message> -->
47 47 <!-- sprintf: -->

```

Рис. 4.16: Текст файла листинга lab7-2.asm

Строка 33: Команда сдвинута на относительный адрес 0000001B, инструкция mov с данными операндами в шестнадцатеричной системе выглядит как: 89C1. Строка 34: Команда сдвинута на относительный адрес 0000001D, инструкция mov с данными операндами в шестнадцатеричной системе выглядит как: BB01000000. Строка 35: Команда сдвинута на относительный адрес 00000022, инструкция mov с данными операндами в шестнадцатеричной системе выглядит как: B804000000.

16. Открыл файл с программой lab7-2.asm и удалил один операнд в инструкции mov (рис. 4.17).

```

Открыть  lab7-2.asm
~/work/arch-pc/lab07
1 %include 'in_out.asm'
2
3 section .data
4     msg1 db 'Введите B: ',0h
5     msg2 db "Наибольшее число: ",0h
6     A dd '20'
7     C dd '50'
8 section .bss
9     max resb 10
10    B resb 10
11 section .text
12    global _start
13 _start:
14    mov eax, ; Удалил операнд B здесь
15    call sprint
16    mov ecx,B
17    mov edx,10
18    call sread
19    mov eax,B
20    call atoi ; Вызов подпрограммы перевода символа в число
21    mov [B],eax ; запись преобразованного числа в 'B'
22    mov ecx,[A] ; 'ecx = A'
23    mov [max],ecx ; 'max = A'
24    cmp ecx,[C] ; Сравниваем 'A' и 'C'
25    jg check_B ; если 'A>C', то переход на метку 'check_B',
26    mov ecx,[C] ; иначе 'ecx = C'
27    mov [max],ecx ; 'max = C'
28    check_B:
29    mov eax,max
30    call atoi ; Вызов подпрограммы перевода символа в число
31    mov [max],eax ; запись преобразованного числа в 'max'
32    mov ecx,[max]
33    cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'
34    jg fin ; если 'max(A,C)>B', то переход на 'fin',
35    mov ecx,[B] ; иначе 'ecx = B'
36    mov [max],ecx
37 fin:
38    mov eax, msg2
39    call sprint ; Вывод сообщения 'Наибольшее число: '
40    mov eax,[max]
41    call iprintLF ; Вывод 'max(A,B,C)'
42    call quit ; Выход

```

Рис. 4.17: Изменённый текст программы

17. Выполнил трансляцию с получением файла листинга рис. 4.18).

```

avperegudov@dk8n76 ~/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
avperegudov@dk8n76 ~/work/arch-pc/lab07 $

```

Рис. 4.18: Результат трансляции

При ошибки в исходном тексте программы трансляция не произойдёт, но файл листинга будет получен. В случае если возникает ошибка в тексте программы, в файле листинга эта ошибка будет отмечена. Также будет написана причина ошибки.

18. Открыл в редакторе gedit полученный файл листинга (рис. 4.19).

```

182 7 00000039 35300000 C dd '50'
183 8 section .bss
184 9 00000000 <res Ah> max resb 10
185 10 0000000A <res Ah> B resb 10
186 11 section .text
187 12 global _start
188 13 _start:
189 14 mov eax, ; Удалил операнд B здесь
190 14 ***** error: invalid combination of opcode and operands
191 15 000000E8 E822FFFFFF call sprint
192 16 000000ED B9[0A000000] mov ecx,B
193 17 000000F2 BA0A000000 mov edx,10
194 18 000000F7 E847FFFFFF call sread
195 19 000000FC B8[0A000000] mov eax,B
196 20 00000101 E896FFFFFF call atoi ; Вызов подпрограммы перевода символа в число
197 21 00000106 A3[0A000000] mov [B],eax ; запись преобразованного числа в 'B'
198 22 0000010B 8B0D[35000000] mov ecx,[A] ; 'ecx = A'
199 23 00000111 890D[00000000] mov [max],ecx ; 'max = A'

```

Рис. 4.19: Текст файла листинга lab7-2.asm

19. Создал файл lab7-3.asm (рис. 4.20).

```

avperegudov@dk8n76 ~/work/arch-pc/lab07 $ touch lab7-3.asm
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ ls

```

Рис. 4.20: Создал файл lab7-3.asm

20. Ввёл текст программы для нахождения наименьшей из 3 целочисленных переменных \diamond, \diamond и c (рис. 4.21, 4.22).

Открыть

lab7-3.asm
~/work/arch-pc/lab07

```

1 %include 'in_out.asm'
2 section .data
3     msgA db 'Введите A: ',0h
4     msgB db 'Введите B: ',0h
5     msgC db 'Введите C: ',0h
6     msgM db "Наименьшее число: ",0h
7
8 section .bss
9     max resb 10
10    A resb 10
11    B resb 10
12    C resb 10
13
14 section .text
15     global _start
16
17 _start:
18
19     mov eax,msgA
20     call sprint
21
22     mov edx,10
23
24     mov ecx,A
25     call sread
26
27     mov eax,msgB
28     call sprint
29
30     mov ecx,B
31     call sread
32
33     mov eax,msgC
34     call sprint
35
36     mov ecx,C
37     call sread
38
39     mov eax,A
40     call atoi
41     mov [A],eax
42
43     mov eax,B
44     call atoi
45     mov [B],eax

```

Рис. 4.21: Текст программы

```

Открыть  lab7-3.asm
~\work\arch-pc\lab07
24      mov ecx,A
25      call sread
26
27      mov eax,msgB
28      call sprint
29
30      mov ecx,B
31      call sread
32
33      mov eax,msgC
34      call sprint
35
36      mov ecx,C
37      call sread
38
39      mov eax,A
40      call atoi
41      mov [A],eax
42
43      mov eax,B
44      call atoi
45      mov [B],eax
46
47      mov eax,C
48      call atoi
49      mov [C],eax
50
51      mov ecx, [A]
52      mov [max], ecx
53      cmp ecx, [B]
54      jnl check_c
55      mov ecx, [B]
56      mov [max], ecx
57
58      check_c:
59      cmp ecx, [C]
60      jnl final
61      mov ecx, [C]
62      mov [max], ecx
63
64      final:
65      mov eax, msgM
66      call sprint
67      mov eax,[max]
68      call iprintfLF
69      call quit
70

```

Рис. 4.22: Продолжение текста

21. Создал исполняемый файл и проверил его работу, используя переменные из 20 варианта (рис. 4.23).

```

avperegudov@Study:~/work/study_2023-2024_arh-pc/labs/lab07/report$ ./lab7-3
Введите A: 95
Введите B: 2
Введите C: 61
Наименьшее число: 2

```

Рис. 4.23: Результат работы программы

22. Создал файл lab7-3.asm (рис. 4.24).

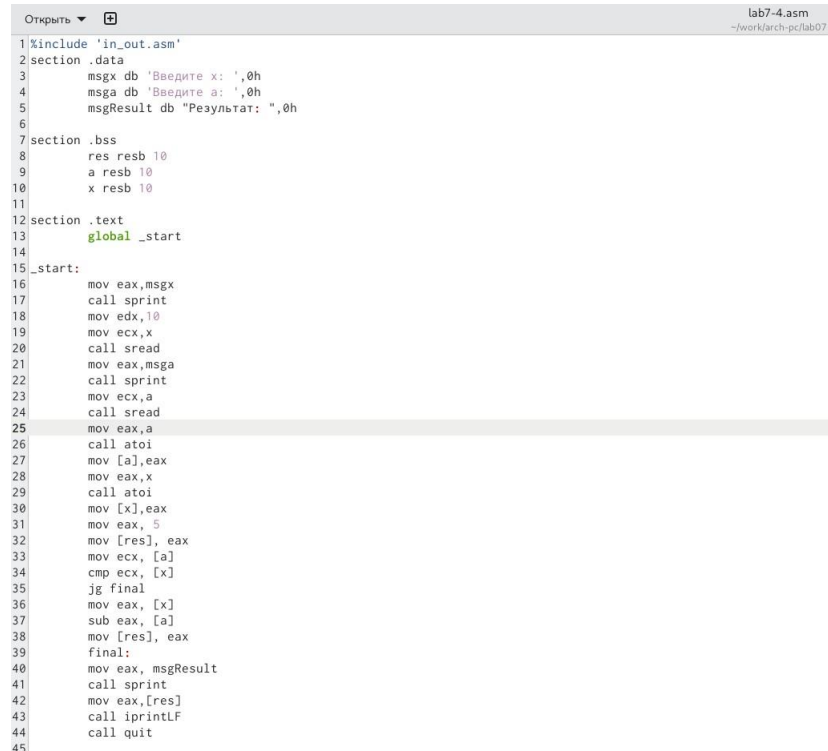
```

avperegudov@dk8n76 ~/work/arch-pc/lab07 $ touch lab7-4.asm

```

Рис. 4.24: Создал файл lab7-3.asm

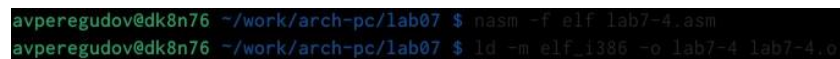
23. Ввёл текст программы, которая для введенных с клавиатуры значений и вычисляет значение функции 20 варианта и выводит результат вычислений (рис. 4.25).



```
1 %include 'in_out.asm'
2 section .data
3     msgx db 'Введите x: ',0h
4     msga db 'Введите a: ',0h
5     msgResult db "Результат: ",0h
6
7 section .bss
8     res resb 10
9     a resb 10
10    x resb 10
11
12 section .text
13     global _start
14
15 _start:
16     mov eax,msgx
17     call sprint
18     mov edx,10
19     mov ecx,x
20     call sread
21     mov eax,msga
22     call sprint
23     mov ecx,a
24     call sread
25     mov eax,a
26     call atoi
27     mov [a],eax
28     mov eax,x
29     call atoi
30     mov [x],eax
31     mov ecx, 5
32     mov [res], eax
33     mov ecx, [a]
34     cmp ecx, [x]
35     jg final
36     mov eax, [x]
37     sub eax, [a]
38     mov [res], eax
39     final:
40     mov eax, msgResult
41     call sprint
42     mov eax,[res]
43     call iprintf
44     call quit
45
```

Рис. 4.25: Текст программы

24. Создал исполняемый файл (рис. 4.26).



```
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ nasm -f elf lab7-4.asm
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-4 lab7-4.o
```

Рис. 4.26: Трансляция и компоновка

25. Запустил программу с параметрами x=1, a=2 (рис. 4.27).

```
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ ./lab7-4
Введите x: 1
Введите a: 2
Результат: 5
avperegudov@dk8n76 ~/work/arch-pc/lab07 $
```

Рис. 4.27: Результат работы программы

26. Запустил программу с параметрами $x=2$, $a=1$ (рис. 4.28).

```
avperegudov@dk8n76 ~/work/arch-pc/lab07 $ ./lab7-4
Введите x: 2
Введите a: 1
Результат: 1
avperegudov@dk8n76 ~/work/arch-pc/lab07 $
```

Рис. 4.28: Результат работы программы

5 Выводы

В данной лабораторной работе были изучены команды условного и безусловного переходов. Были приобретены навыки написания программ с использованием переходов. Также были изучены способ создания, назначение и структура файла листинга.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.