

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра физико-математических и естественных наук

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

дисциплина: Архитектура компьютера

Студент: Попова Алиса Владимировна

Группа: НПИ-03-25

МОСКВА

2025г.

Цель работы

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

Задание 1

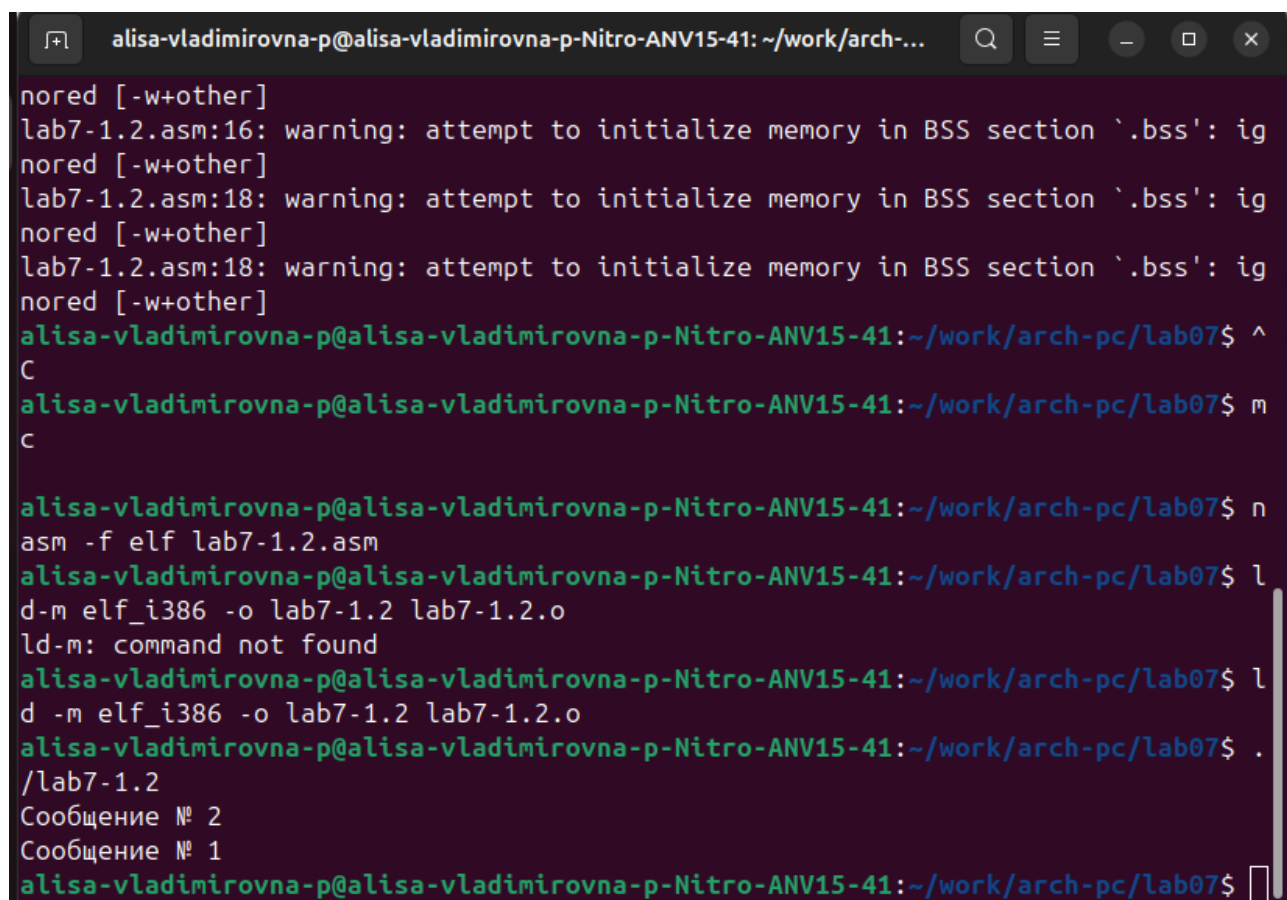
1. Создайте каталог для программ лабораторной работы № 7, перейдите в него и создайте файл lab7-1.asm

```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/arch-...
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~$ mkdir ~/work/arch-pc/lab07
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~$ cd ~/work/arch-pc/lab07
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ touch lab7-1.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ nasm -f elf lab07.asm
nasm: fatal: unable to open input file `lab07.asm' No such file or directory
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$
```

Создан каталог для 7 лабораторной работы, создан файл lab7-1.asm и исполняемый файл, при запуске получен ожидаемый результат.

Задание 1.2

Измените текст программы в соответствии с листингом 7.2. Создайте исполняемый файл и проверьте его работу.



```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/arch-...
nored [-w+other]
lab7-1.2.asm:16: warning: attempt to initialize memory in BSS section `.bss': ig
nored [-w+other]
lab7-1.2.asm:18: warning: attempt to initialize memory in BSS section `.bss': ig
nored [-w+other]
lab7-1.2.asm:18: warning: attempt to initialize memory in BSS section `.bss': ig
nored [-w+other]
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ ^
C
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ m
c

alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ n
asm -f elf lab7-1.2.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ l
d-m elf_i386 -o lab7-1.2 lab7-1.2.o
ld-m: command not found
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ l
d -m elf_i386 -o lab7-1.2 lab7-1.2.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ .
/lab7-1.2
Сообщение № 2
Сообщение № 1
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$
```

Программа отредактирована, создан исполняемый файл получен
удовлетворяющий результат

Задание 2

Создайте файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Внимательно изучите текст программы из листинга 7.3 и введите в lab7-2.asm.

Программа должна:

1. Иметь три переменные: A, B, C

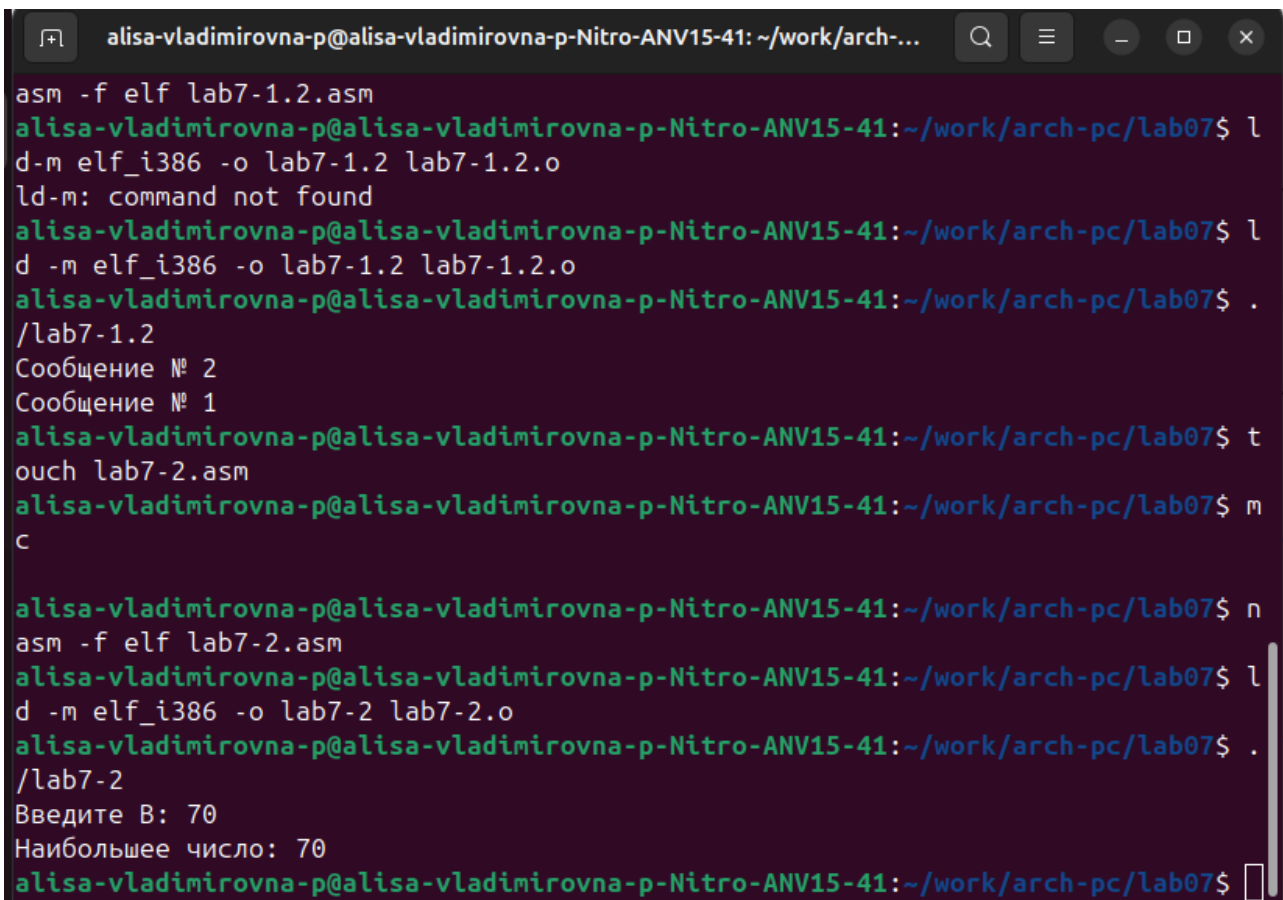
- A и C заданы в программе (жёстко прописаны)
- B вводится с клавиатуры

2. Найти наибольшее из трёх чисел A, B, C

- Сравнить A и C → выбрать большее в max
- Сравнить max(A,C) с B → выбрать наибольшее из них

3. Вывести результат

- Вывести сообщение "Наибольшее число: "
- Вывести само число



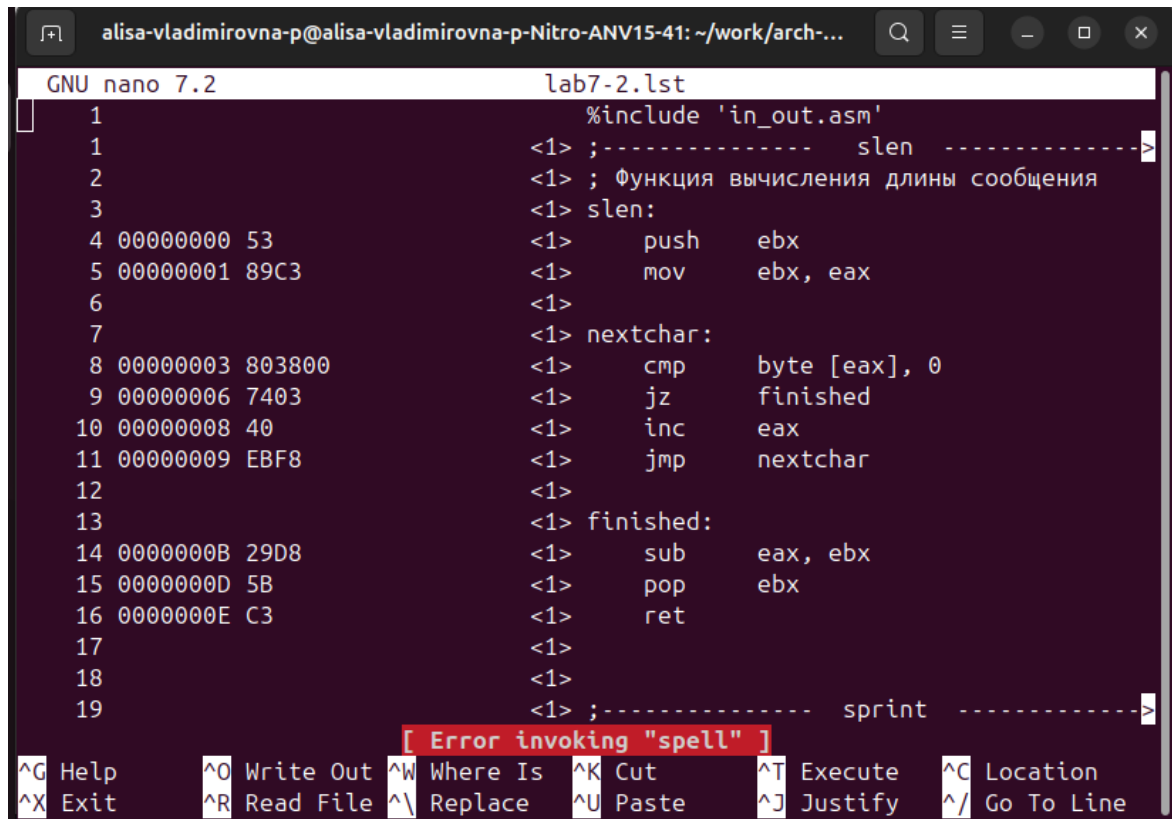
```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/arch-...
asm -f elf lab7-1.2.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ l
d-m elf_i386 -o lab7-1.2 lab7-1.2.o
ld-m: command not found
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ l
d -m elf_i386 -o lab7-1.2 lab7-1.2.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ .
/lab7-1.2
Сообщение № 2
Сообщение № 1
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ t
ouch lab7-2.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ m
c
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ n
asm -f elf lab7-2.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ l
d -m elf_i386 -o lab7-2 lab7-2.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ .
/lab7-2
Введите B: 70
Наибольшее число: 70
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$
```

Создан файл lab7-2.asm, запущен, но удовлетворяющий результат получен

Задание 3

Создайте файл листинга для программы из файла lab7-2.asm

Откройте файл листинга lab7-2.lst с помощью любого текстового редактора,



```
GNU nano 7.2 lab7-2.lst
1 %include 'in_out.asm'
1 <1> ;----- slen ----->
2 <1> ; Функция вычисления длины сообщения
3 <1> slen:
4 00000000 53 <1> push ebx
5 00000001 89C3 <1> mov ebx, eax
6 <1>
7 <1> nextchar:
8 00000003 803800 <1> cmp byte [eax], 0
9 00000006 7403 <1> jz finished
10 00000008 40 <1> inc eax
11 00000009 EBF8 <1> jmp nextchar
12 <1>
13 <1> finished:
14 0000000B 29D8 <1> sub eax, ebx
15 0000000D 5B <1> pop ebx
16 0000000E C3 <1> ret
17 <1>
18 <1>
19 <1> ;----- sprint ----->
[ Error invoking "spell" ]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line
```

Внимательно ознакомиться с его форматом и содержанием. Подробно объяснить содержание

одной из трёх строк файла листинга по выбору.

Строка 4

4 00000000 53 <1> push ebx

- 4 : порядковый номер строки в файле листинга.
- 00000000: адрес (смещение) в байтах от начала текущей секции (скорее всего, секции .text). Это значит, что данная инструкция находится в самом начале секции кода.
- 53 : машинный код в шестнадцатеричном формате. Код 53 соответствует инструкции push ebx в архитектуре x86.
- <1> : уровень вложенности (depth level) макроса или повторения. Цифра 1 может означать, что эта строка находится внутри первого уровня макроподстановки или цикла %rep. Если бы было <0>, это означало бы основной код.
- push ebx : исходный текст программы на ассемблере. Инструкция сохраняет значение регистра ebx в стек.

Строка 5

5 00000001 89C3 <1> mov ebx, eax

5 : номер строки в листинге.

- 00000001 : адрес этой инструкции. Она следует сразу за предыдущей (разница в 1 байт: push ebx заняла 1 байт 53).

- 89C3 : машинный код инструкции mov ebx, eax. В x86:

89 :опкод для mov r/m32, r32

C3 : ModR/M байт, указывающий на ebx как приёмник и eax как источник.

- <1> : снова уровень вложенности 1 (такой же, как у предыдущей строки).
- mov ebx, eax : исходная инструкция: копирует значение из регистра eax в регистр ebx.

Строка 6

6 <1>

- 6 : номер строки.
- Адрес отсутствует : потому что в этой строке нет машинного кода.
- Машинный код отсутствует : потому что это пустая строка в исходном коде.
- <1> : уровень вложенности сохраняется (1).
- Исходный текст отсутствует : это просто пустая строка в исходном файле .asm, добавленная для читаемости.

Откройте файл с программой lab7-2.asm и в любой инструкции с двумя операндами удалить один операнд. Выполните трансляцию с получением файла листинга

```
mov [max], eax
fin:
; Вывод результата
mov eax, 
call sprint
mov eax, [max]
call iprintLF
call quit
```

```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ n
c
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ n
asm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:53: error: invalid combination of opcode and operands
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$
```

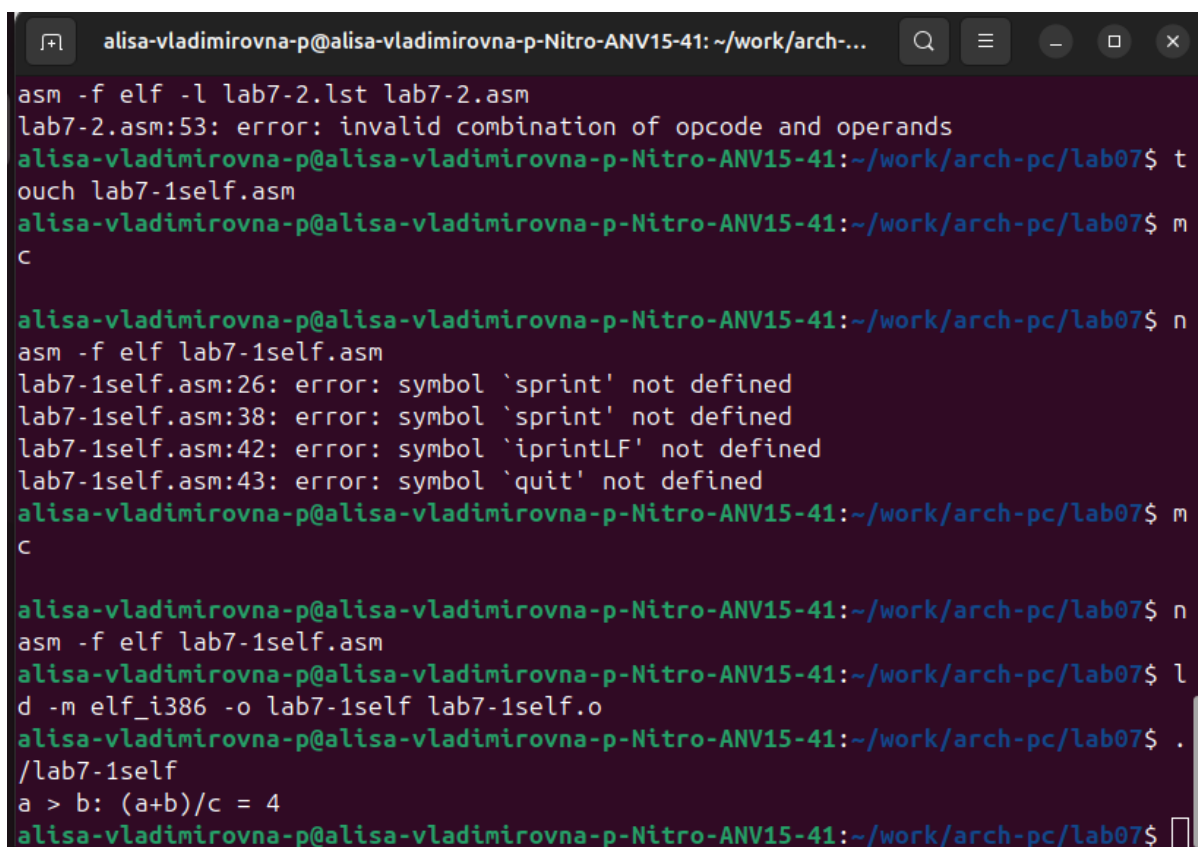
удалён операнд msg2, выполнена трансляция и запущен файл листинга. В результате удаления операнда программа не смогла запуститься и выдало ошибку.

Задание для самостоятельной работы

1. Напишите программу нахождения наименьшей из 3 целочисленных переменных a, b и c . Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу.

2. Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции $f(x)$ и выводит результат вычислений. Вид функции $f(x)$ выбрать из таблицы 7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу для значений x и a из 7.6

1 Создан файл lab7-1self.asm, куда была записана нахождения наименьшей из 3



```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/arch-...
asm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:53: error: invalid combination of opcode and operands
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ touch lab7-1self.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ mc
c

alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ nasm -f elf lab7-1self.asm
lab7-1self.asm:26: error: symbol `sprintf' not defined
lab7-1self.asm:38: error: symbol `sprintf' not defined
lab7-1self.asm:42: error: symbol `iprintLF' not defined
lab7-1self.asm:43: error: symbol `quit' not defined
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ mc
c

alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ nasm -f elf lab7-1self.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1self lab7-1self.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ ./lab7-1self
a > b: (a+b)/c = 4
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$
```

целочисленных переменных a, b и C . Вариант 12 $A=99, B=29, C=26$.

2 Создан файл lab7-2self.asm, куда была записана программа которая для введенных с клавиатуры значений x и a вычисляет значение заданной

Номер варианта	Выражение для $f(x)$	(x_1, a_1)	(x_2, a_2)
8	$\begin{cases} 3a, & a < 3 \\ x + 1, & a \geq 3 \end{cases}$	(1;4)	(1;2)
9	$\begin{cases} a + x, & x \leq a \\ a, & x > a \end{cases}$	(5;7)	(6;4)
10	$\begin{cases} x - 2, & x > 2 \\ 3a, & x \leq 2 \end{cases}$	(3;0)	(1;2)
11	$\begin{cases} 4a, & x = 0 \\ 4a + x, & x \neq 0 \end{cases}$	(0;3)	(1;2)
12	$\begin{cases} ax, & x < 5 \\ x - 5, & x \geq 5 \end{cases}$	(3;7)	(6;4)

```

alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/arch-...
asm -f elf lab7-1self.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ l
d -m elf_i386 -o lab7-1self lab7-1self.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ .
/lab7-1self
a > b: (a+b)/c = 4
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ t
ouch lab7-2self.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ m
c

alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ n
asm -f elf lab7-2self.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ l
d -m elf_386 -o lab7-2self lab7-2self.o
ld: unrecognized emulation mode: elf_386
Supported emulations: elf_x86_64 elf32_x86_64 elf_i386 elf_iamcu i386pep i386pe
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ l
d -m elf_i386 -o lab7-2self lab7-2self.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$ .
/lab7-2self
Для (x1=3, a1=7): f(x) = 21
Для (x2=6, a2=4): f(x) = 1
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab07$

```

функции $f(x)$ и выводит результат вычислений. Использован вариант 12

Вывод

В ходе выполнения лабораторной работы №7 были изучены команды безусловного (jmp) и условных переходов (je, jg, jl и др.), приобретены практические навыки написания программ с ветвлениями на ассемблере NASM, а также исследована структура и назначение файла листинга. Были успешно разработаны и отлажены программы, реализующие алгоритмы с условными переходами, включая вычисление выражений в зависимости от условий и кусочно-заданных функций. Кроме того, был выполнен анализ файла листинга, что позволило глубже понять процесс трансляции исходного кода в машинные инструкции. Все поставленные цели работы достигнуты в полном объёме.

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/Learning-bash-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, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ-Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. 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.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).