

# **Отчёт по лабораторной работе №5**

**Дисциплина: архитектура компьютера**

Незами Ахмад Белал

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>6</b>
<b>2</b>	<b>Задание</b>	<b>7</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
3.0.1	Изучение Midnight Commander . . . . .	8
3.0.2	Структура программы на языке ассемблера NASM . . . . .	11
3.0.3	Подключение внешнего файла . . . . .	13
<b>4</b>	<b>Выполнение заданий для самостоятельной работы</b>	<b>17</b>
4.0.1	1 . . . . .	17
4.0.2	2 . . . . .	20
<b>5</b>	<b>Выводы</b>	<b>22</b>
<b>6</b>	<b>Список литературы</b>	<b>23</b>

## Список иллюстраций

3.1	Открытие Midnight Commander . . . . .	8
3.2	Перехожу в каталог ~/work/arch-рс, используя файловый менеджер mc . . . . .	9
3.3	С помощью функциональной клавиши F7 создаю каталог lab05 . .	10
3.4	В строке ввода прописываю команду touch lab5-1.asm, чтобы со- здать файл, в котором буду работать . . . . .	11
3.5	С помощью функциональной клавиши F3 открываю файл для про- смotra, чтобы проверить, содержит ли файл текст программы . .	12
3.6	Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу . . . . .	12
3.7	Скачиваю файл in_out.asm со страницы курса в ТУИС. Он сохра- нился в каталог “Загрузки” . . . . .	13
3.8	С помощью функциональной клавиши F6 переместил файл in_out.asm из каталога Загрузки в созданный каталог lab05 . . . .	13
3.9	С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне mc прописываю имя lab5-2.asm для копии файла . . . . .	14
3.10	. . . . .	14
3.11	Изменяю содержимое файла lab5-2.asm во встроенном редакторе nano, чтобы в программе использовались подпрограммы из внеш- него файла in_out.asm. . . . .	15
3.12	Транслирую текст программы файла в объектный файл командой nasm -f elf lab5-2.asm. Создался объектный файл lab5-2.o. Выполняю компоновку объектного файла с помощью команды ld -m elf_i386 -o lab5-2 lab5-2.o Создался исполняемый файл lab5-2. Запускаю исполняемый файл . . . . .	15
3.13	Открываю файл lab5-2.asm для редактирования в nano функцио- нальной клавишей F4. Изменяю в нем подпрограмму sprintLF на sprint. Сохраняю изменения и открываю файл для просмотра, что- бы проверить сохранение действий . . . . .	16
3.14	Снова транслирую файл, выполняю компоновку созданного объ- ектного файла, запускаю новый исполняемый файл . . . . .	16
4.1	Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5 . . . . .	17
4.2	Проверка . . . . .	18

4.3	С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку . . . . .	19
4.4	Создаю объектный файл lab5-1-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-1-1, запускаю полученный исполняемый файл. Программа запрашивает ввод, ввожу свои ФИО, далее программа выводит введенные мною данные . . . . .	20
4.5	Создаю копию файла lab5-2.asm с именем lab5-2-2.asm с помощью функциональной клавиши F5 . . . . .	20
4.6	С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку . . . . .	21
4.7	Создаю объектный файл lab5-2-2.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab6-2-2, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО, далее программа выводит введенные мною данные . . . . .	21

## Список таблиц

# 1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера `mov` и `int`.

## 2 Задание

1. Открыть Midnight Commander.
2. Создать папку lab05, где будут храниться файлы лабораторной работы №5.
3. Создать файл «lab5-1.asm», ввести текст программы. Оттранслировать текст программы, выполнить компоновку объектного файла и запустить получившийся исполняемый файл.
4. Скачать с ТУИС файл «in\_out.asm» и переместить его в каталог lab05.
5. Скопировать файл «lab5-1.asm» с именем «lab5-2.asm» и исправить текст программы так чтобы использовались программы из внешнего файла «in\_out.asm».
6. Создать исполняемый файл и проверить его работу.
7. Создать копию файла «lab5-1.asm». Внести изменения в программу (без использования внешнего файла «in\_out.asm»), так чтобы она работала по определённому алгоритму.
8. Создать копию файла «lab5-2.asm». Также исправить текст программы, но уже с использованием подпрограмм из внешнего файла «in\_out.asm», так чтобы она работала по определённому алгоритму.
9. Создать исполняемые файлы и проверить их работу.

## 3 Выполнение лабораторной работы

### 3.0.1 Изучение Midnight Commander

Открываем Midnight Commander с помощью команды 'mc'

A terminal window with a dark background. The prompt is [nikta1382@fedora ~]\$ and the command mc is being entered, followed by a white cursor block.

```
[nikta1382@fedora ~]$ mc
```

Рис. 3.1: Открытие Midnight Commander



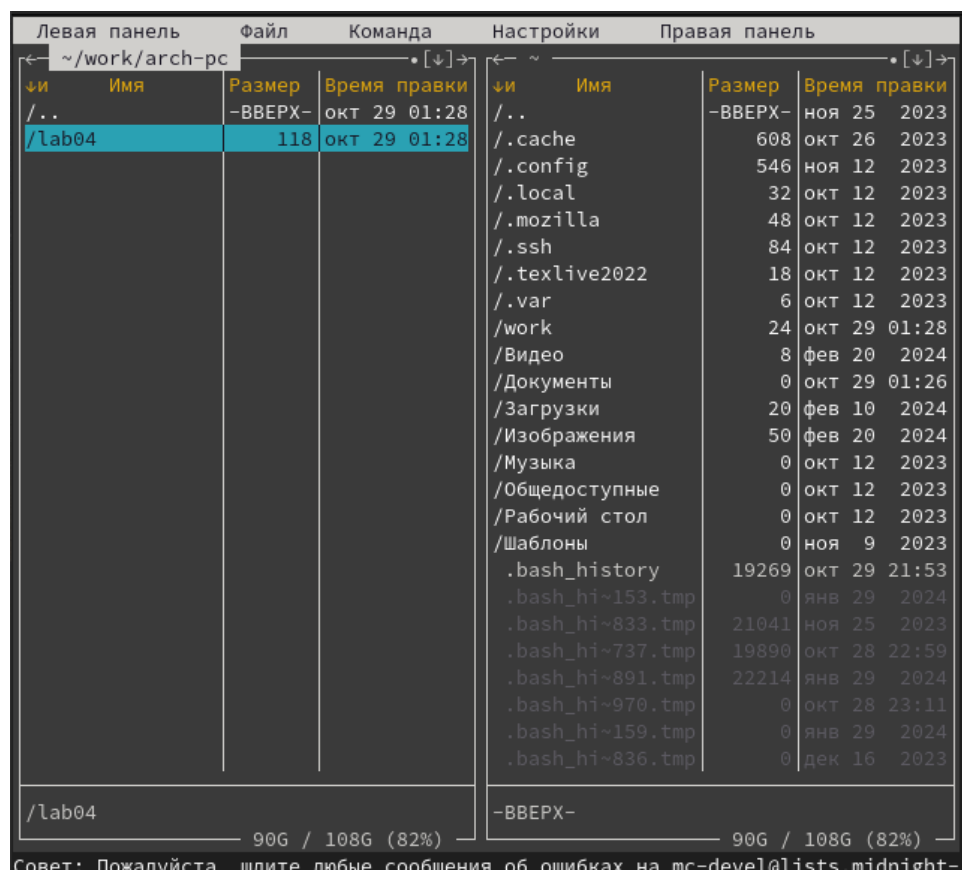


Рис. 3.2: Перехожу в каталог ~/work/arch-pc, используя файловый менеджер mc

Левая панель			Файл	Команда	Настройки	Правая панель		
← ~ /work/arch-pc				• [↕] →	← ~		• [↕] →	
Имя	Размер	Время правки	Имя	Размер	Время правки	Имя	Размер	Время правки
/..	-ВВЕРХ-	окт 29 01:28	/..	-ВВЕРХ-	ноя 25 2023	/..	-ВВЕРХ-	ноя 25 2023
/lab04	118	окт 29 01:28	/.cache	608	окт 26 2023	/.cache	608	окт 26 2023
/lab05	0	окт 29 22:01	/.config	546	ноя 12 2023	/.config	546	ноя 12 2023
			/.local	32	окт 12 2023	/.local	32	окт 12 2023
			/.mozilla	48	окт 12 2023	/.mozilla	48	окт 12 2023
			/.ssh	84	окт 12 2023	/.ssh	84	окт 12 2023
			/.texlive2022	18	окт 12 2023	/.texlive2022	18	окт 12 2023
			/.var	6	окт 12 2023	/.var	6	окт 12 2023
			/work	24	окт 29 01:28	/work	24	окт 29 01:28
			/Видео	8	фев 20 2024	/Видео	8	фев 20 2024
			/Документы	0	окт 29 01:26	/Документы	0	окт 29 01:26
			/Загрузки	20	фев 10 2024	/Загрузки	20	фев 10 2024
			/Изображения	50	фев 20 2024	/Изображения	50	фев 20 2024
			/Музыка	0	окт 12 2023	/Музыка	0	окт 12 2023
			/Общедоступные	0	окт 12 2023	/Общедоступные	0	окт 12 2023
			/Рабочий стол	0	окт 12 2023	/Рабочий стол	0	окт 12 2023
			/Шаблоны	0	ноя 9 2023	/Шаблоны	0	ноя 9 2023
			.bash_history	19269	окт 29 21:53	.bash_history	19269	окт 29 21:53
			.bash_hi~153.tmp	0	янв 29 2024	.bash_hi~153.tmp	0	янв 29 2024
			.bash_hi~833.tmp	21041	ноя 25 2023	.bash_hi~833.tmp	21041	ноя 25 2023
			.bash_hi~737.tmp	19890	окт 28 22:59	.bash_hi~737.tmp	19890	окт 28 22:59
			.bash_hi~891.tmp	22214	янв 29 2024	.bash_hi~891.tmp	22214	янв 29 2024
			.bash_hi~970.tmp	0	окт 28 23:11	.bash_hi~970.tmp	0	окт 28 23:11
			.bash_hi~159.tmp	0	янв 29 2024	.bash_hi~159.tmp	0	янв 29 2024
			.bash_hi~836.tmp	0	дек 16 2023	.bash_hi~836.tmp	0	дек 16 2023
/lab05			-ВВЕРХ-					
90G / 108G (82%)			90G / 108G (82%)			90G / 108G (82%)		
Совет: Пожалуйста, шлите любые сообщения об ошибках на mc-devel@lists.midnightbsd.org								

Рис. 3.3: С помощью функциональной клавиши F7 создаю каталог lab05

Перехожу в созданный каталог

Левая панель			Команда	Правая панель		
Файл				Настройки		
Имя	Размер	Время правки		Имя	Размер	Время правки
lab5-1.asm	0	окт 29 22:03		..	-ВВЕРХ-	ноя 25 2023
				/.cache	608	окт 26 2023
				/.config	546	ноя 12 2023
				/.local	32	окт 12 2023
				/.mozilla	48	окт 12 2023
				/.ssh	84	окт 12 2023
				/.texlive2022	18	окт 12 2023
				/.var	6	окт 12 2023
				/work	24	окт 29 01:28
				/Видео	8	фев 20 2024
				/Документы	0	окт 29 01:26
				/Загрузки	20	фев 10 2024
				/Изображения	50	фев 20 2024
				/Музыка	0	окт 12 2023
				/Общедоступные	0	окт 12 2023
				/Рабочий стол	0	окт 12 2023
				/Шаблоны	0	ноя 9 2023
				.bash_history	19269	окт 29 21:53
				.bash_history~153.tmp	0	янв 29 2024
				.bash_history~833.tmp	21041	ноя 25 2023
				.bash_history~737.tmp	19890	окт 28 22:59
				.bash_history~891.tmp	22214	янв 29 2024
				.bash_history~970.tmp	0	окт 28 23:11
				.bash_history~159.tmp	0	янв 29 2024
				.bash_history~836.tmp	0	дек 16 2023
lab5-1.asm				-ВВЕРХ-		

Рис. 3.4: В строке ввода прописываю команду `touch lab5-1.asm`, чтобы создать файл, в котором буду работать

### 3.0.2 Структура программы на языке ассемблера NASM

С помощью функциональной клавиши F4 открываю созданный файл для редактирования в редакторе nano. Ввожу в файл код программы для запроса строки у пользователя. Далее выхожу из файла (Ctrl+X), сохраняя изменения (Y, Enter).

```

SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программ
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
mov eax, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ;Descriptor файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

```

Рис. 3.5: С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-1.asm`. Создался объектный файл `lab5-1.o`. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-1 lab5-1.o`. Создался исполняемый файл `lab5-1`.

```

[nikta1382@fedora lab05]$ nasm -f elf lab5-1.asm
[nikta1382@fedora lab05]$ ld -m elf_i386 -o lab5-1 lab5-1.o
[nikta1382@fedora lab05]$ ./lab5-1
Введите строку:
Хаджилари Гешлаг Никта
[nikta1382@fedora lab05]$

```

Рис. 3.6: Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу

### 3.0.3 Подключение внешнего файла

Левая панель		Файл		Команда		Настройки		Правая панель	
← ~/work/arch-pc/lab05						← ~/Загрузки			
↓И	Имя	Размер	Время правки	↓И	Имя	Размер	Время правки	↓И	Имя
/..		-ВВЕРХ-	окт 29 22:01	/..		-ВВЕРХ-	окт 29 22:09	/..	
*lab5-1		8744	окт 29 22:12	in_out.asm		3942	окт 28 2023		
lab5-1.asm		1312	окт 29 22:09						
lab5-1.o		752	окт 29 22:11						

Рис. 3.7: Скачиваю файл in\_out.asm со страницы курса в ТУИС. Он сохранился в каталог “Загрузки”

Левая панель			Файл	Команда	Настройки	Правая панель					
← ~/work/arch-pc/lab05			• [↓]			← ~/Загрузки			• [↓]		
↓ Имя	Размер	Время правки	↓ Имя	Размер	Время правки	↓ Имя	Размер	Время правки			
/..	-ВВЕРХ-	окт 29 22:01	/..	-ВВЕРХ-	окт 29 22:09						
in_out.asm	3942	окт 28 2023									
*lab5-1	8744	окт 29 22:12									
lab5-1.asm	1312	окт 29 22:09									
lab5-1.o	752	окт 29 22:11									

Рис. 3.8: С помощью функциональной клавиши F6 переместил файл in\_out.asm из каталога Загрузки в созданный каталог lab05

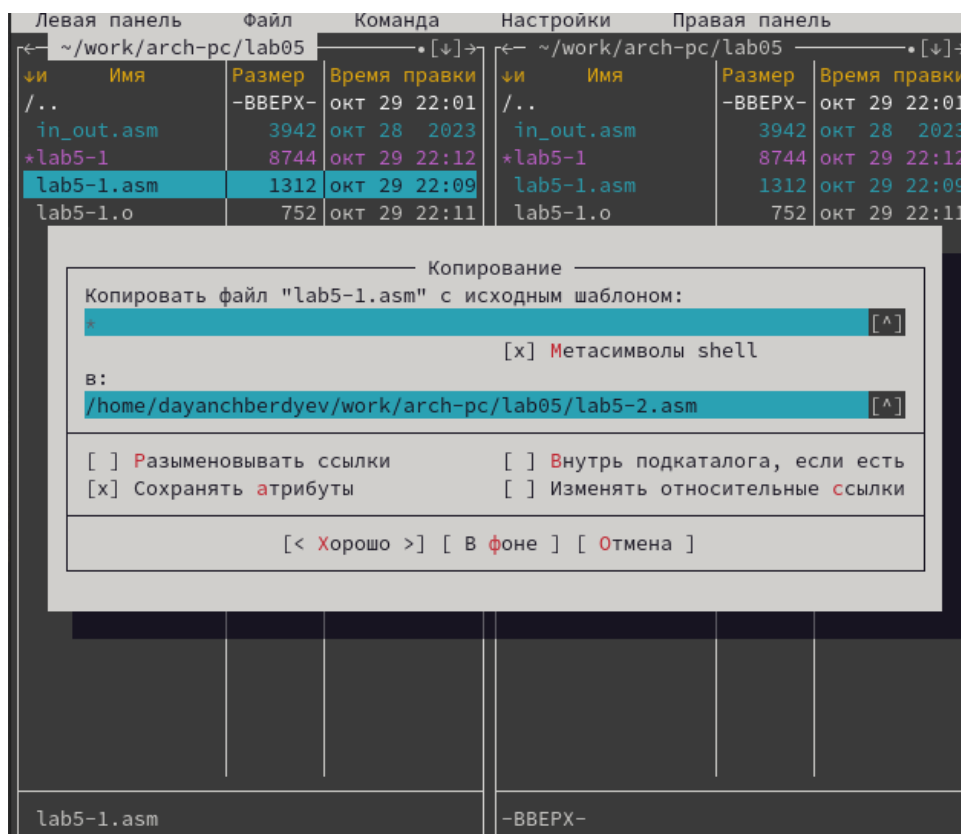


Рис. 3.9: С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне тс прописываю имя lab5-2.asm для копии файла

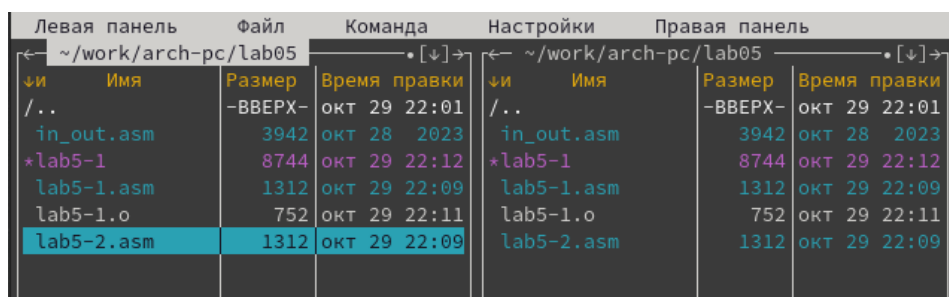
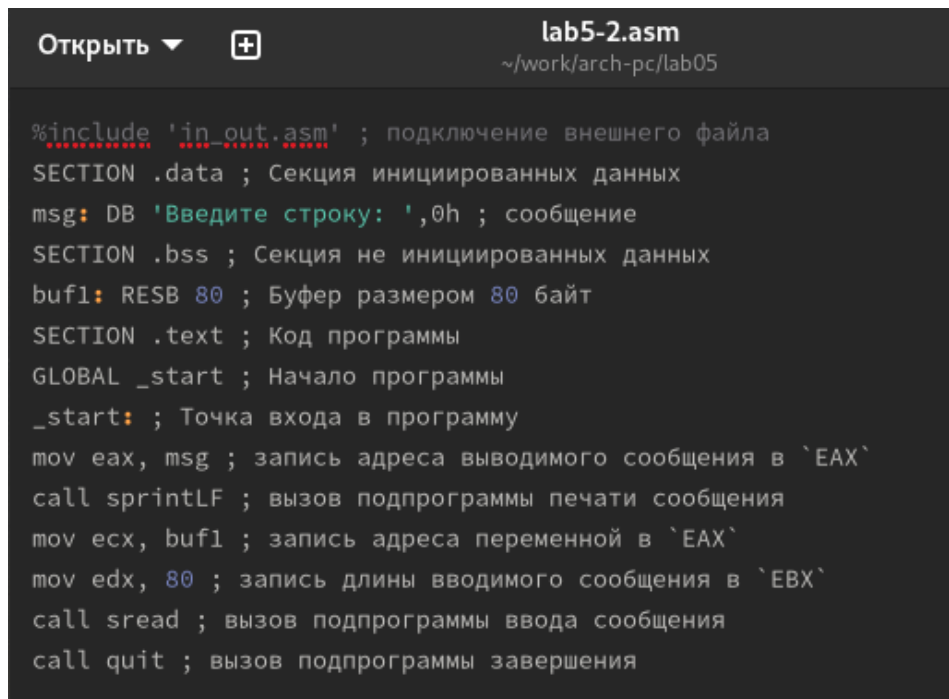


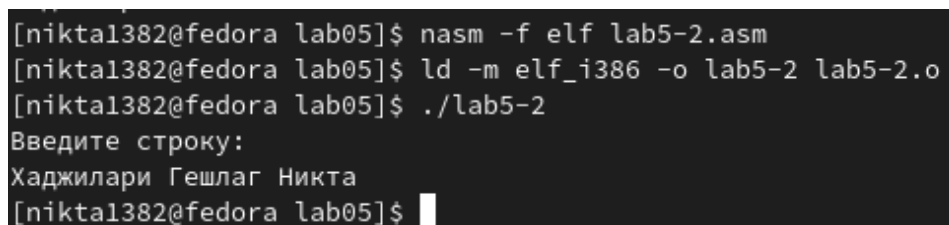
Рис. 3.10:



```
Открыть ▾ + lab5-2.asm
~/work/arch-pc/lab05

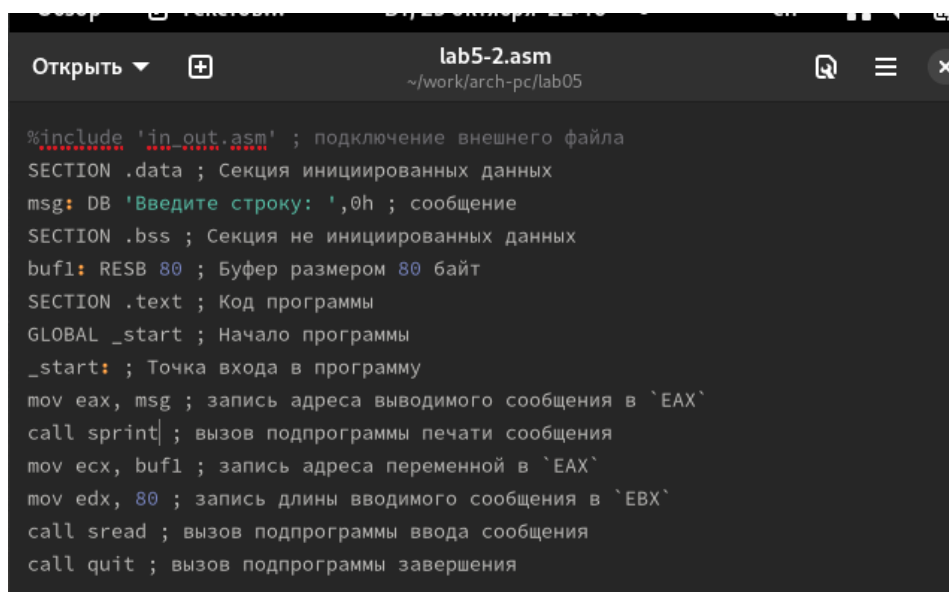
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprintf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 3.11: Изменяю содержимое файла lab5-2.asm во встроенном редакторе nano, чтобы в программе использовались подпрограммы из внешнего файла in\_out.asm.



```
[nikta1382@fedora lab05]$ nasm -f elf lab5-2.asm
[nikta1382@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[nikta1382@fedora lab05]$ ./lab5-2
Введите строку:
Хаджилари Гешлаг Никта
[nikta1382@fedora lab05]$
```

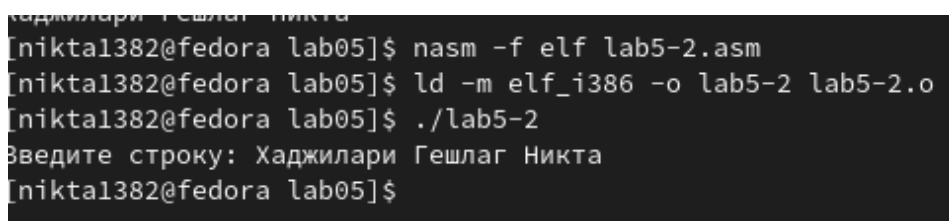
Рис. 3.12: Транслирую текст программы файла в объектный файл командой nasm -f elf lab5-2.asm. Создался объектный файл lab5-2.o. Выполняю компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab5-2 lab5-2.o Создался исполняемый файл lab5-2. Запускаю исполняемый файл



```
lab5-2.asm
~/work/arch-pc/lab05

#include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprint ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 3.13: Открываю файл lab5-2.asm для редактирования в nano функциональной клавишей F4. Изменяю в нем подпрограмму printf на sprint. Сохраняю изменения и открываю файл для просмотра, чтобы проверить сохранение действий



```
[nikta1382@fedora lab05]$ nasm -f elf lab5-2.asm
[nikta1382@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[nikta1382@fedora lab05]$ ./lab5-2
Введите строку: Хаджилари Гешлаг Никта
[nikta1382@fedora lab05]$
```

Рис. 3.14: Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл

Разница между первым исполняемым файлом lab5-2 и вторым lab5-2 в том, что запуск первого запрашивает ввод с новой строки, а программа, которая выполняется при запуске второго, запрашивает ввод без переноса на новую строку, потому что в этом заключается различие между подпрограммами printf и sprint.



## 4 Выполнение заданий для самостоятельной работы

### 4.0.1 1

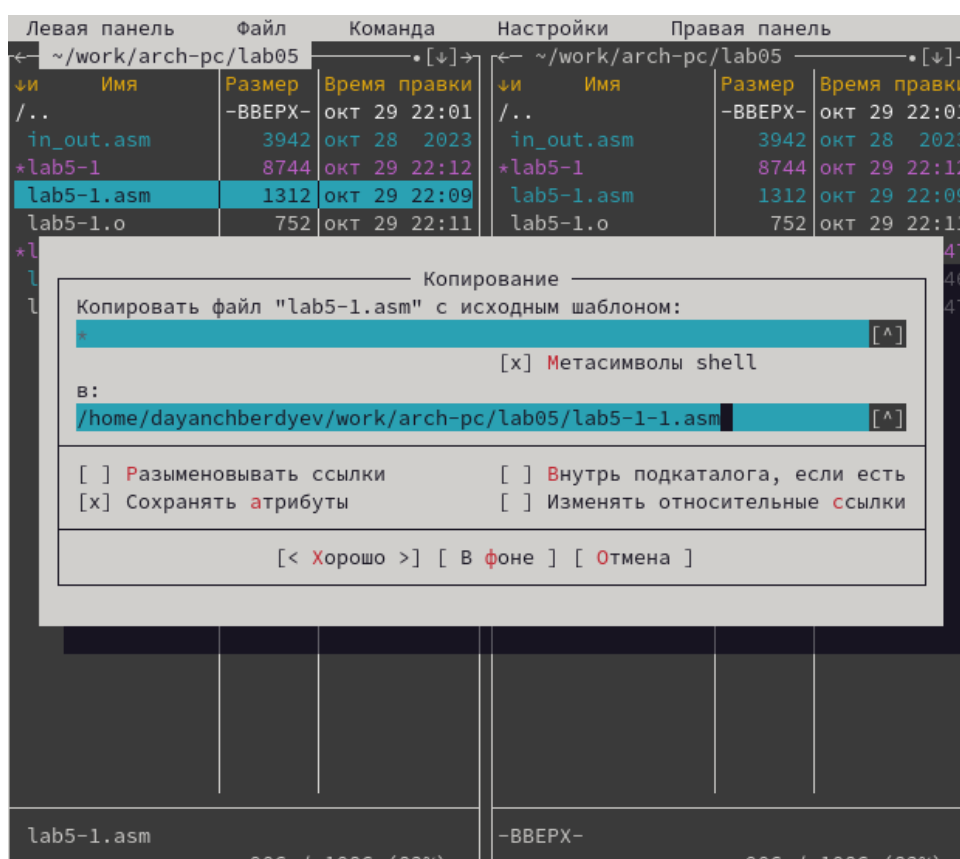
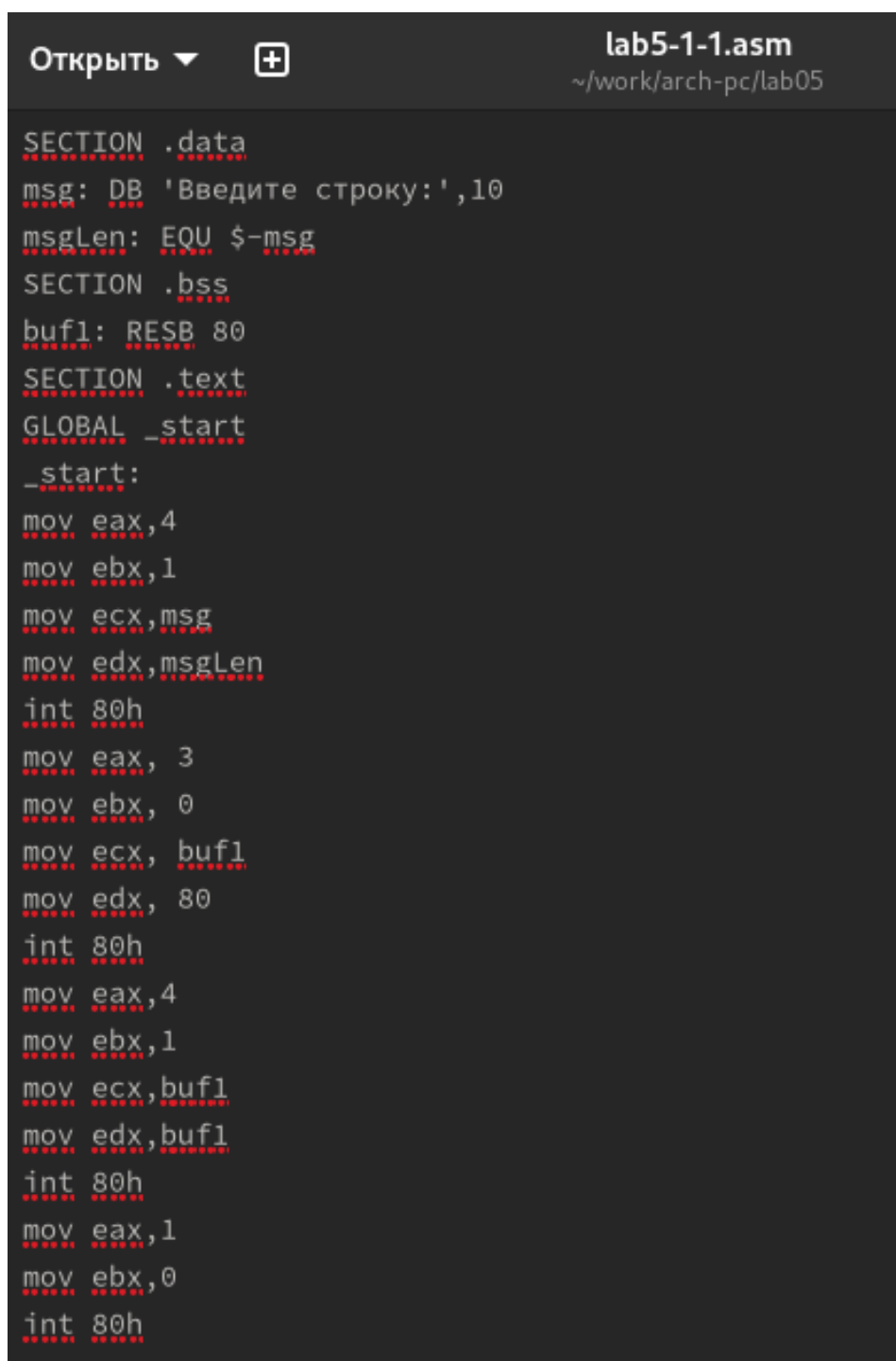



Рис. 4.1: Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5

Левая панель			Файл	Команда	Настройки	Правая панель		
← ~/work/arch-pc/lab05				•[↓]→		← ~/work/arch-pc/lab05		
Имя	Размер	Время правки				Имя	Размер	Время правки
../	-ВВЕРХ-	окт 29 22:01				../	-ВВЕРХ-	окт 29 22:01
in_out.asm	3942	окт 28 2023				in_out.asm	3942	окт 28 2023
*lab5-1	8744	окт 29 22:12				*lab5-1	8744	окт 29 22:12
lab5-1-1.asm	1312	окт 29 22:09				lab5-1-1.asm	1312	окт 29 22:09
lab5-1.asm	1312	окт 29 22:09				lab5-1.asm	1312	окт 29 22:09
lab5-1.o	752	окт 29 22:11				lab5-1.o	752	окт 29 22:11
*lab5-2	9092	окт 29 22:47				*lab5-2	9092	окт 29 22:47
lab5-2.asm	963	окт 29 22:46				lab5-2.asm	963	окт 29 22:46
lab5-2.o	1312	окт 29 22:47				lab5-2.o	1312	окт 29 22:47
lab5-1-1.asm			-ВВЕРХ-					
90G / 108G (82%)			90G / 108G (82%)					

Рис. 4.2: Проверка



```
Открыть ▼  lab5-1-1.asm
~/work/arch-pc/lab05

SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,buf1
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 4.3: С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку

```

[nikta1382@fedora lab05]$ nasm -f elf lab5-1-1.asm
[nikta1382@fedora lab05]$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
[nikta1382@fedora lab05]$ ./lab5-1-1
Введите строку:
Хаджилари Гешлаг Никта
Хаджилари Гешлаг Никта
[nikta1382@fedora lab05]$

```

Рис. 4.4: Создаю объектный файл lab5-1-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-1-1, запускаю полученный исполняемый файл. Программа запрашивает ввод, ввожу свои ФИО, далее программа выводит введенные мною данные

## 4.0.2 2

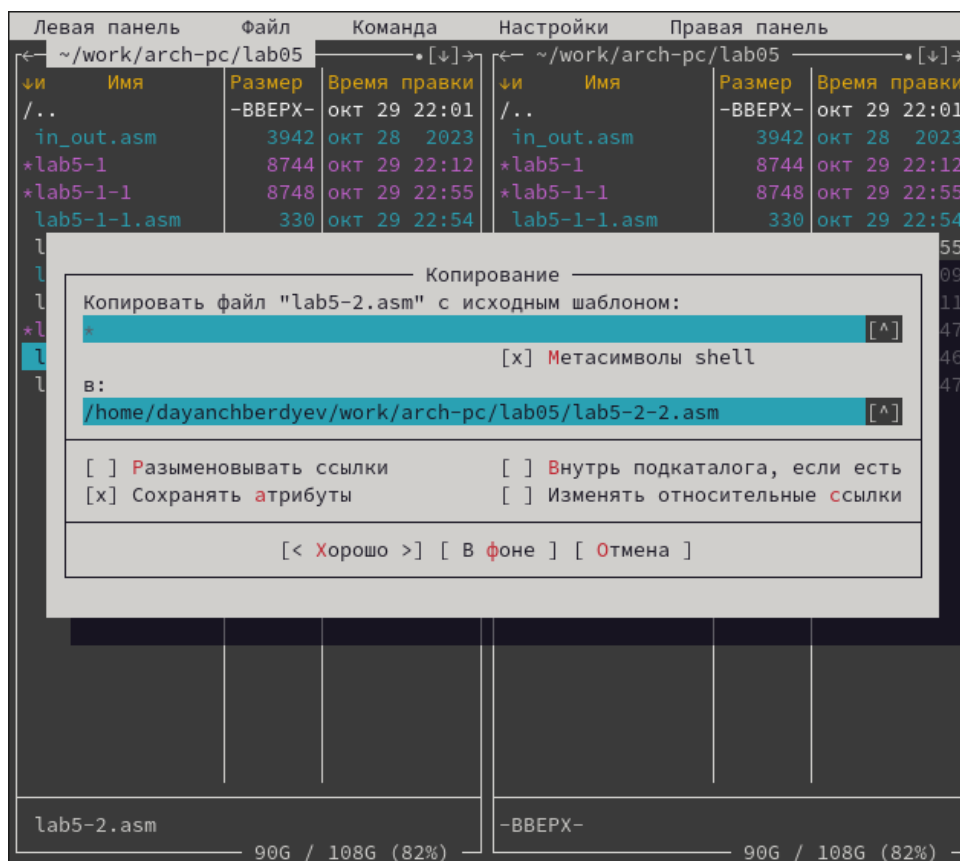
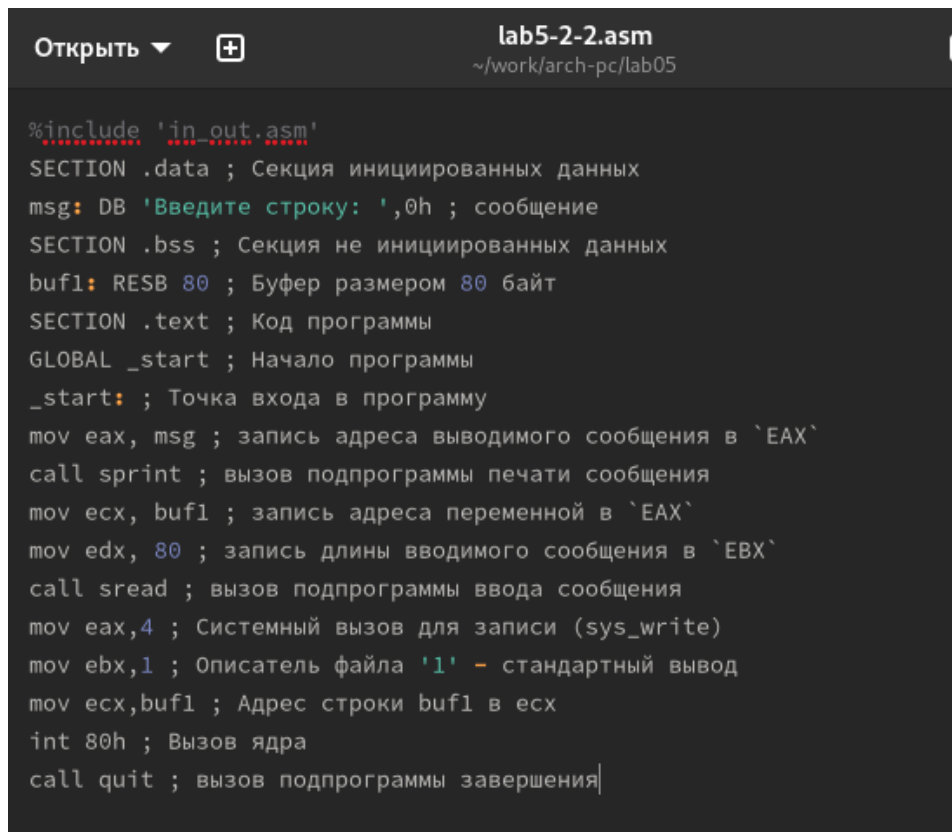


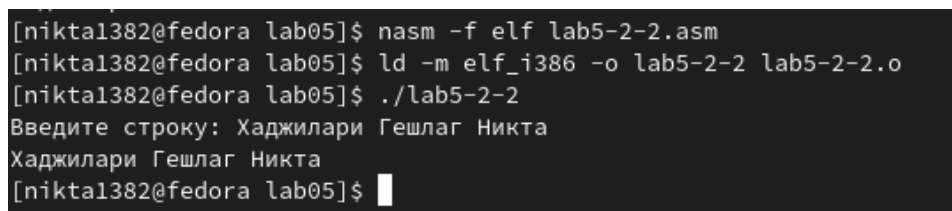
Рис. 4.5: Создаю копию файла lab5-2.asm с именем lab5-2-2.asm с помощью функциональной клавиши F5



```
Открыть ▾ + lab5-2-2.asm
~/work/arch-pc/lab05

%include 'in_out.asm'
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprint ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
mov eax, 4 ; Системный вызов для записи (sys_write)
mov ebx, 1 ; Описатель файла '1' - стандартный вывод
mov ecx, buf1 ; Адрес строки buf1 в ecx
int 80h ; Вызов ядра
call quit ; вызов подпрограммы завершения
```

Рис. 4.6: С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку



```
[nikta1382@fedora lab05]$ nasm -f elf lab5-2-2.asm
[nikta1382@fedora lab05]$ ld -m elf_i386 -o lab5-2-2 lab5-2-2.o
[nikta1382@fedora lab05]$ ./lab5-2-2
Введите строку: Хаджилари Гешлаг Никта
Хаджилари Гешлаг Никта
[nikta1382@fedora lab05]$
```

Рис. 4.7: Создаю объектный файл lab5-2-2.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab6-2-2, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО, далее программа выводит введенные мною данные

## 5 Выводы

При выполнении данной лабораторной работы я приобрел практические навыки работы в Midnight Commander, а также освоил инструкции языка ассемблера `mov` и `int`.

## 6 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](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).