

Отчёт по лабораторной работе №5

дисциплина: Архитектура компьютеров

Маслова Анна Павловна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение заданий для самостоятельной работы	17
4	Выводы	23
	Список литературы	24

Список иллюстраций

2.1	Midnight Commander	6
2.2	Каталог ~/work/arch-pc	6
2.3	Каталог lab05	7
2.4	Строка ввода	7
2.5	Созданный файл	8
2.6	Редактор mcedit	8
2.7	Текст программы вывода сообщения на экран и ввода строки . . .	9
2.8	Содержимое файла lab5-1.asm	10
2.9	Транслирование, компоновка lab5-1.asm и запуск программы . .	10
2.10	Загруженный файл in_out.asm	11
2.11	Копирование файла in_out.asm в каталог lab05	11
2.12	Файл in_out.asm в каталоге lab05	12
2.13	Перемещение файла lab5-1.asm	13
2.14	Файл lab5-1.asm заменён на lab5-2.asm	13
2.15	Копирование файла lab5-1.asm	14
2.16	Копия файла lab5-1.asm с именем lab5-2.asm	14
2.17	Текст программы в файле lab5-2.asm	15
2.18	Запуск lab5-2	15
2.19	Замена sprintLF на sprint	16
2.20	Запуск lab5-2 с исправленной подпрограммой	16
3.1	Созданный файл lab5-1c.asm	17
3.2	Текст программы в файле lab5-1c.asm	19
3.3	Запуск lab5-1c	20
3.4	Созданный файл lab5-2c.asm	20
3.5	Текст программы в файле lab5-2c.asm	22
3.6	Запуск lab5-2c	22

Список таблиц

1 Цель работы

Приобрести навыки работы в *Midnight Commander* на практике. Освоить инструкции языка ассемблера *mov* и *int*.

2 Выполнение лабораторной работы

С помощью команды `mc` откроем *Midnight Commander* (рис. 2.1).

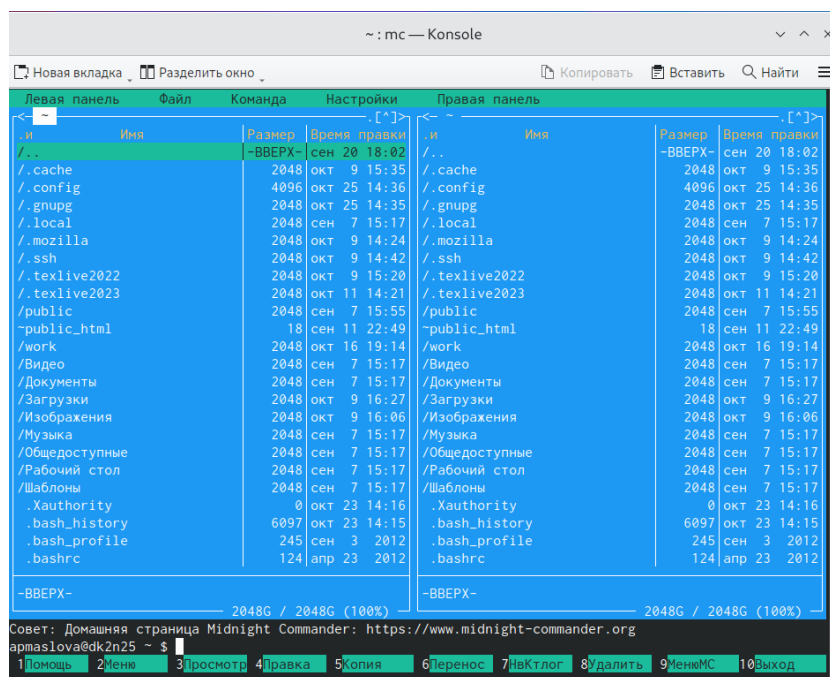


Рис. 2.1: Midnight Commander

Затем с помощью клавиатуры переходим в каталог `~/work/arch-pc` (рис. 2.2).

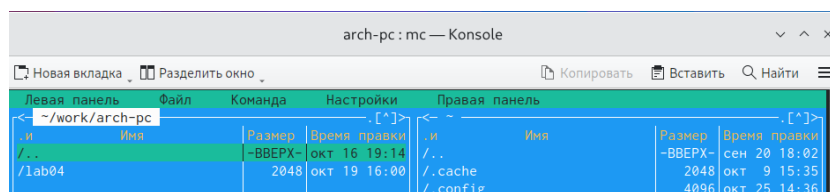


Рис. 2.2: Каталог `~/work/arch-pc`

В этом каталоге с помощью клавиши F7 создадим папку *lab05* и перейдём в неё (рис. 2.3).

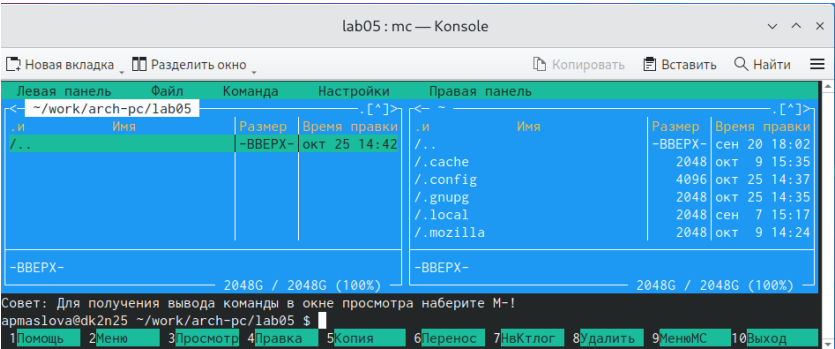


Рис. 2.3: Каталог lab05

Далее ниже строке ввода вводим команду *touch*, с помощью которой создадим файл *lab5-1.asm* (рис. 2.4).

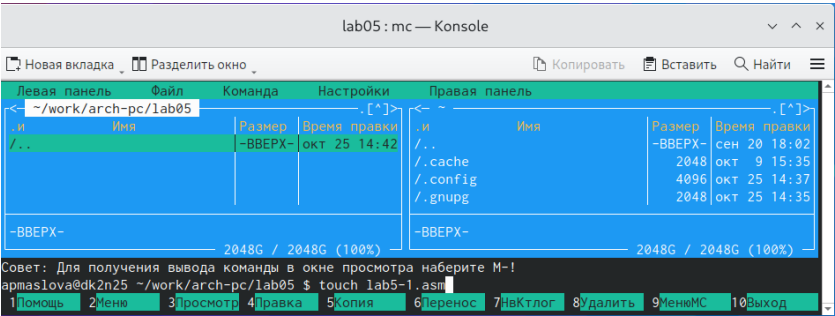


Рис. 2.4: Строка ввода

На рис. 2.5 видим, что файл создан:

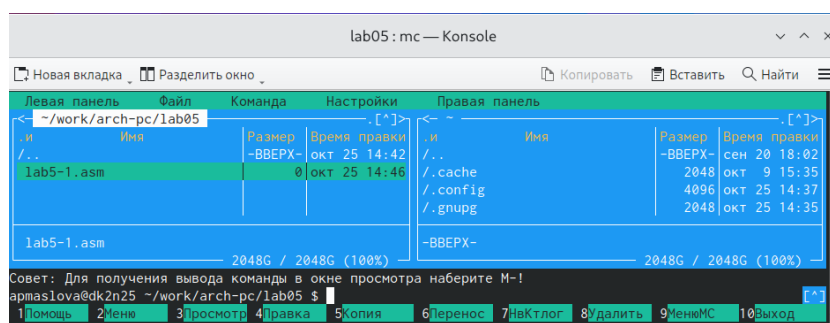


Рис. 2.5: Созданный файл

С помощью клавиши F4 откроем созданный файл для редактирования (рис. 2.6).

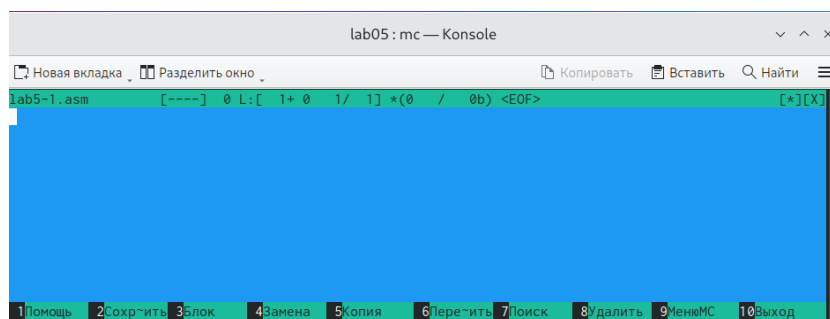
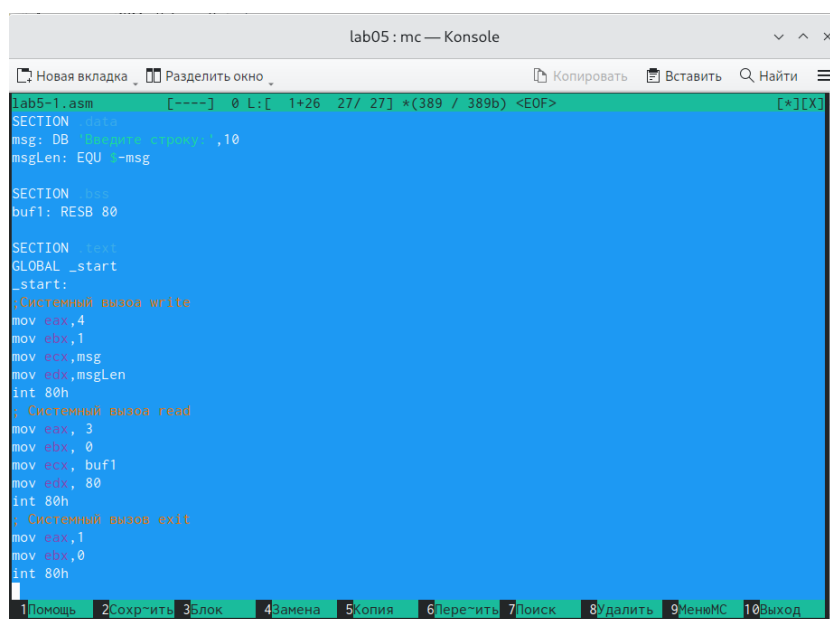


Рис. 2.6: Редактор mcedit

Видим, что открылся редактор mcedit.

Введём текст программы вывода сообщения “Введите строку” на экран и ввода строки с клавиатуры (рис. 2.7).



```
lab5-1.asm [-----] 0 L: [ 1+26 27/ 27] *(389 / 389b) <EOF> [*][X]
SECTION .data
msg: DB "Введите строку",10
msgLen: EQU $-msg

SECTION .resb
buf1: RESB 80

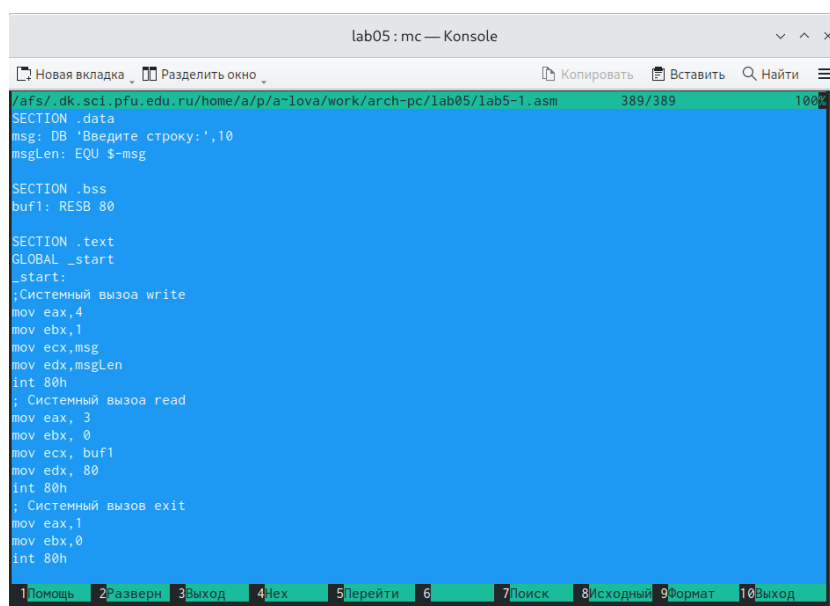
SECTION .text
GLOBAL _start
_start:
; Системный вызов write
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
; Системный вызов read
mov eax,3
mov ebx,0
mov ecx,buf1
mov edx,80
int 80h
; Системный вызов exit
mov eax,1
mov ebx,0
int 80h

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перезагрузить 7Поиск 8Удалить 9МенюМС 10Выход
```

Рис. 2.7: Текст программы вывода сообщения на экран и ввода строки

Сохранили изменения в файле с помощью функциональной клавиши F2 и закрыли mcedit, используя клавишу F10.

Нажмём клавишу F3, выделив файл *lab5-1.asm*, и проверим, что в нём содержится текст написанной программы (рис. 2.8).



```
lab05: mc — Konsole
/afs/.dk.sci.pfu.edu.ru/home/a/p/a-lova/work/arch-pc/lab05/lab5-1.asm 389/389 100%
SECTION .data
msg: DB "Введите строку:",10
msgLen: EQU $-msg

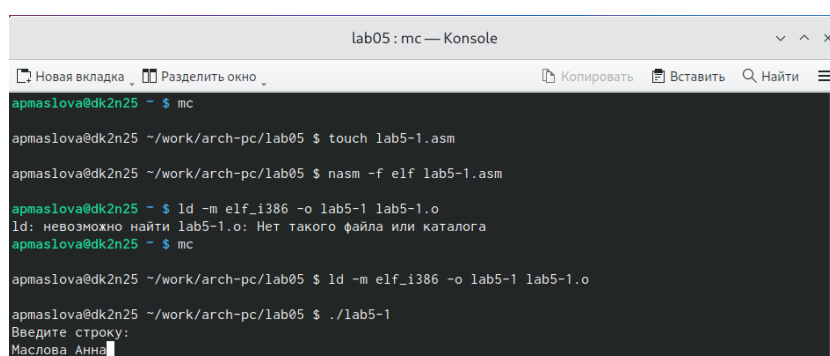
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:
; Системный вызов write
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
; Системный вызов read
mov eax,3
mov ebx,0
mov ecx,buf1
mov edx,80
int 80h
; Системный вызов exit
mov eax,1
mov ebx,0
int 80h

1Помощь 2Разверн 3Выход 4Нех 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход
```

Рис. 2.8: Содержимое файла lab5-1.asm

Теперь оттранслируем текст программы в объектный файл, а затем объектный файл скомпилируем и запустим получившийся исполняемый файл (рис. 2.9).



```
lab05: mc — Konsole
армаслова@dk2n25 ~ $ mc
армаслова@dk2n25 ~/work/arch-pc/lab05 $ touch lab5-1.asm
армаслова@dk2n25 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
армаслова@dk2n25 ~ $ ld -m elf_i386 -o lab5-1 lab5-1.o
ld: невозможно найти lab5-1.o: Нет такого файла или каталога
армаслова@dk2n25 ~ $ mc
армаслова@dk2n25 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
армаслова@dk2n25 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Маслова Анна
```

Рис. 2.9: Транслирование, компоновка lab5-1.asm и запуск программы

Как мы видим, программа работает: на экран выводится сообщение “Введите строку:” и считывается введенная строка (в нашем случае имя и фамилия - Маслова Анна).

Теперь создадим программу, выполняющую такую же функцию, но с помощью внешнего файла. Для этого с ТУИС скачиваем файл *in_out.asm* (рис. 2.10).

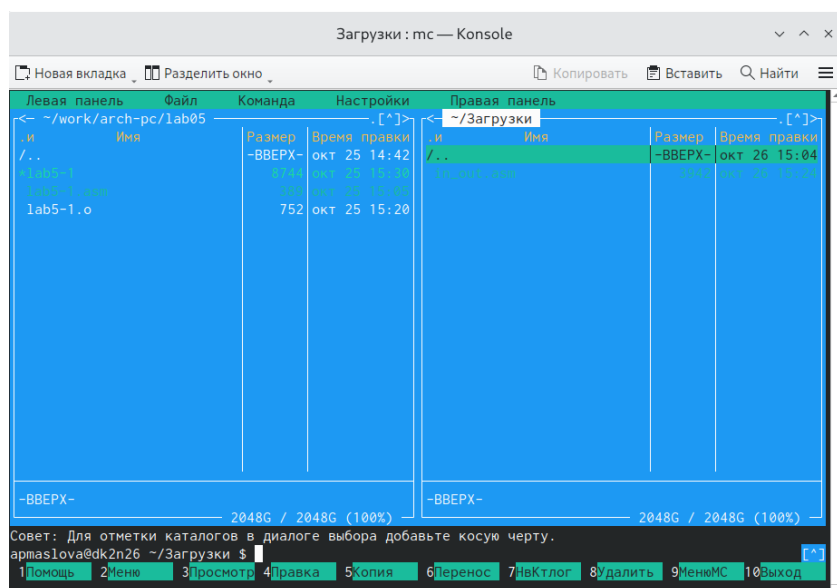


Рис. 2.10: Загруженный файл in_out.asm

Загруженный файл *in_out.asm* помещён в каталог *Загрузки*. Чтобы применять его в программе *lab5-1.asm* он должен лежать в том же каталоге *~/work/arch-pc/lab05*. Скопируем этот файл в нужный каталог с помощью функциональной клавиши F5 (рис. 2.11).

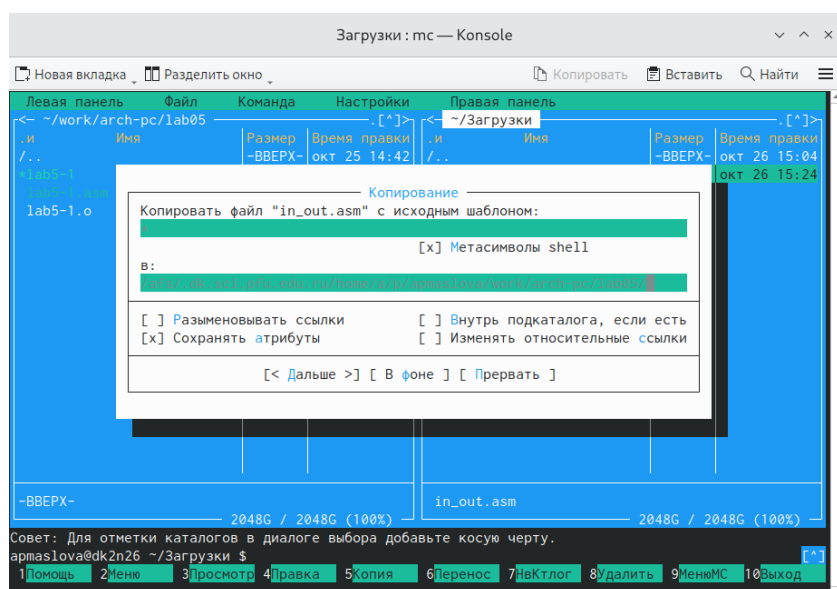


Рис. 2.11: Копирование файла in_out.asm в каталог lab05

На рис. 2.12 можем видеть, что файл *in_out.asm* содержится в каталоге *~/work/arch-pc/lab05*.

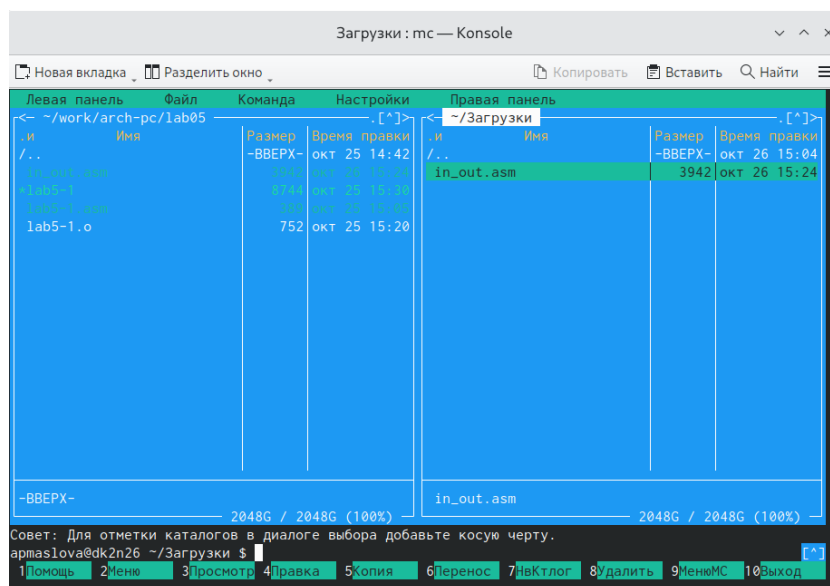


Рис. 2.12: Файл *in_out.asm* в каталоге *lab05*

Далее нам нужно создать файл *lab5-2.asm*. Выделим файл *lab5-1.asm*. С помощью клавиши F6 мы можем переместить (то есть заменить) файл *lab5-1.asm* на файл *lab5-2.asm* (рис. 2.13 , рис. 2.14).

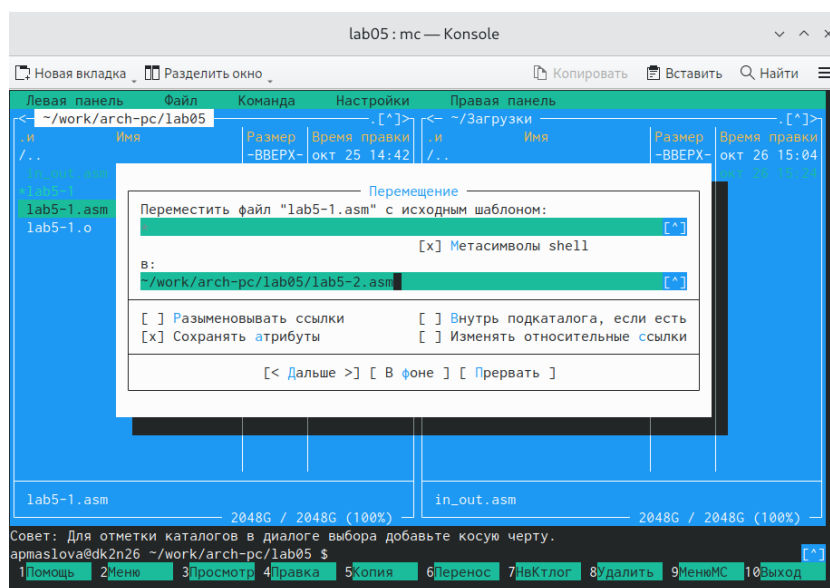


Рис. 2.13: Перемещение файла lab5-1.asm

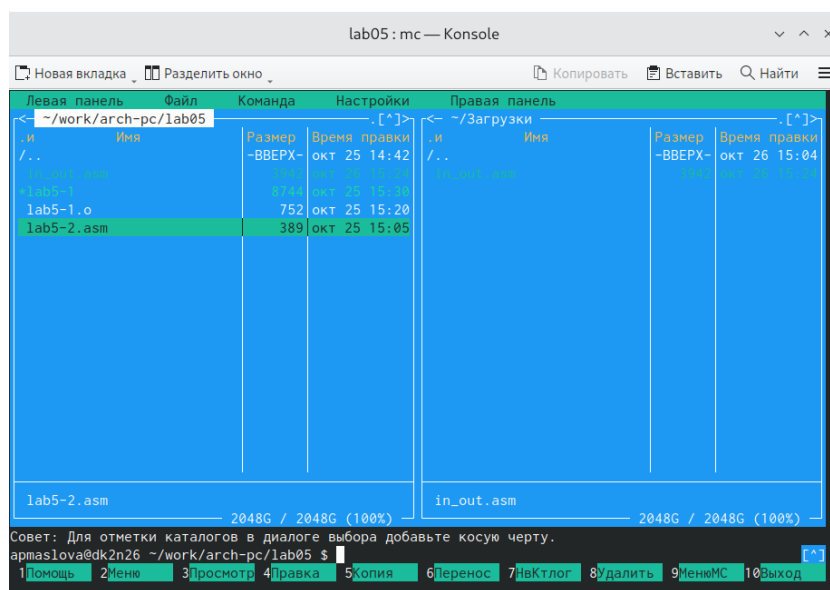


Рис. 2.14: Файл lab5-1.asm заменён на lab5-2.asm

Однако файл *lab5-1.asm* нам пригодится в ходе выполнения дальнейших заданий лабораторной работы, поэтому вместо перемещения клавишей F6 файл *lab5-1.asm* мы скопируем в этот же каталог с помощью клавиши F5 и зададим ему имя “*lab5-2.asm*” (рис. 2.15 , рис. 2.16).

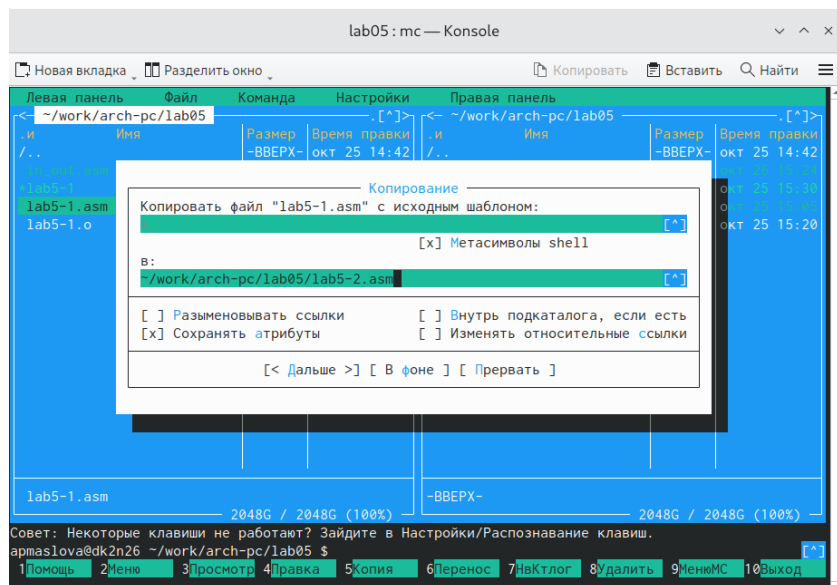
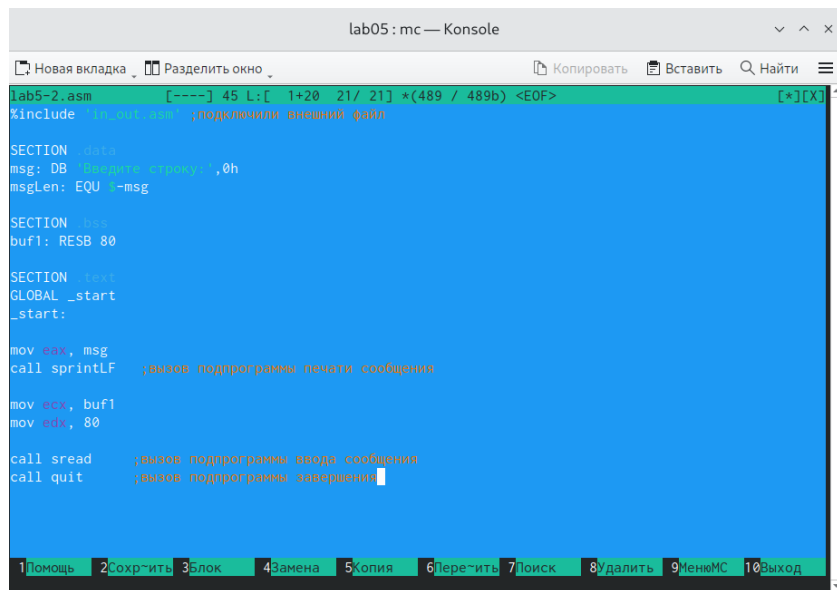


Рис. 2.15: Копирование файла lab5-1.asm



Рис. 2.16: Копия файла lab5-1.asm с именем lab5-2.asm

Откроем созданный файл в редакторе и исправим текст программы в нём с использованием подпрограмм из загруженного внешнего файла *in_out.asm*, а именно - подпрограмм *sprintLF*, *sread*, *guit* (рис. 2.17).



```
lab5-2.asm [----] 45 L: [ 1+20 21/ 21] *(489 / 489b) <EOF> [*][X]
#include "lab5-2.h" ;подключили внешний файл

SECTION .data
msg: DB "Введите строку: ",0h
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

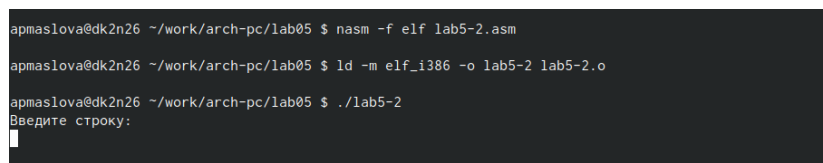
mov eax, msg
call sprintf ;вызов подпрограммы печати сообщения

mov ecx, buf1
mov edx, 80

call sread ;вызов подпрограммы ввода сообщения
call quit ;вызов подпрограммы завершения
```

Рис. 2.17: Текст программы в файле lab5-2.asm

Создадим исполняемый файл и проверим его работу (рис. 2.18).

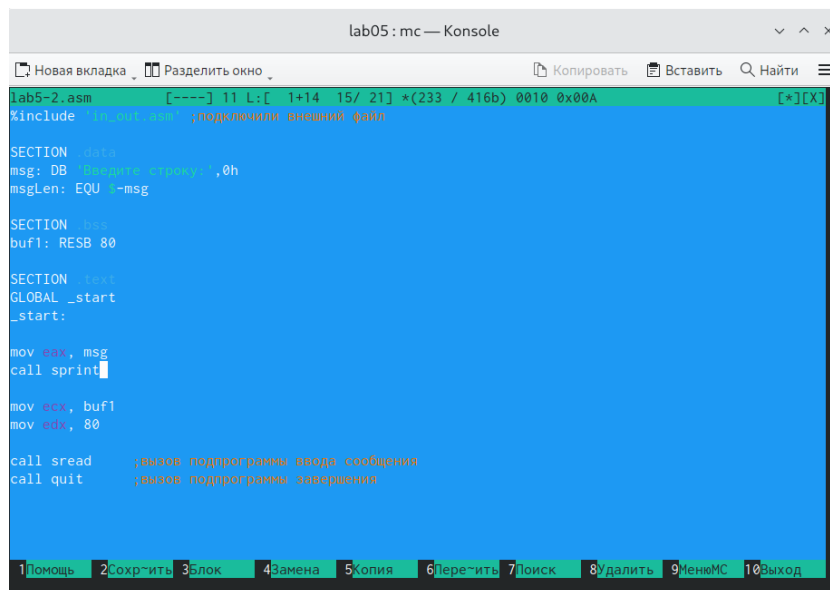


```
армаслова@dk2n26 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
армаслова@dk2n26 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
армаслова@dk2n26 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
```

Рис. 2.18: Запуск lab5-2

Как мы видим, программа выполняет свою функцию.

А теперь снова откроем в редакторе файл *lab5-2.asm* и заменим в нём подпрограмму *sprintf* на *sprint* (рис. 2.19).



```
lab5-2.asm [----] 11 L: [ 1+14 15/ 21] *(233 / 416b) 0010 0x00A [*][X]
#include "in-out.asm";подключили внешний файл

SECTION .data
msg: DB "Введите строку:",0h
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprint

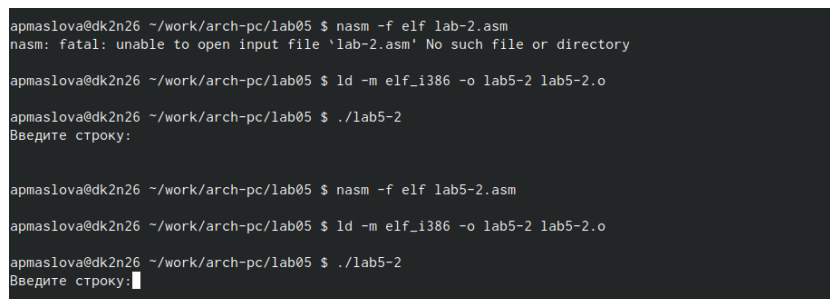
mov ecx, buf1
mov edx, 80

call sread ;вызов подпрограммы ввода сообщения
call quit ;вызов подпрограммы завершения

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перенести 7Поиск 8Удалить 9МенюМС 10Выход
```

Рис. 2.19: Замена `sprintLF` на `sprint`

Создадим исполняемый файл и запустим его (рис. 2.20).



```
armaslova@dk2n26 ~/work/arch-pc/lab05 $ nasm -f elf lab-2.asm
nasm: fatal: unable to open input file 'lab-2.asm' No such file or directory

armaslova@dk2n26 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o

armaslova@dk2n26 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:

armaslova@dk2n26 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm

armaslova@dk2n26 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o

armaslova@dk2n26 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: 
```

Рис. 2.20: Запуск `lab5-2` с исправленной подпрограммой

Файл запустился. Можно заметить, что подпрограмма `sprint` в отличие от подпрограммы `sprintLF` не переносит строку. То есть в первом случае мы вводили нашу строку “Маслова Анна” на следующей строке после сообщения “Введите строку:”, а во втором случае - в той же.

3 Выполнение заданий для самостоятельной работы

Создадим копию файла *lab5-1.asm* и зададим имя новому файлу “*lab5-1c.asm*”. Для этого используем клавишу F5 (рис. 3.1).

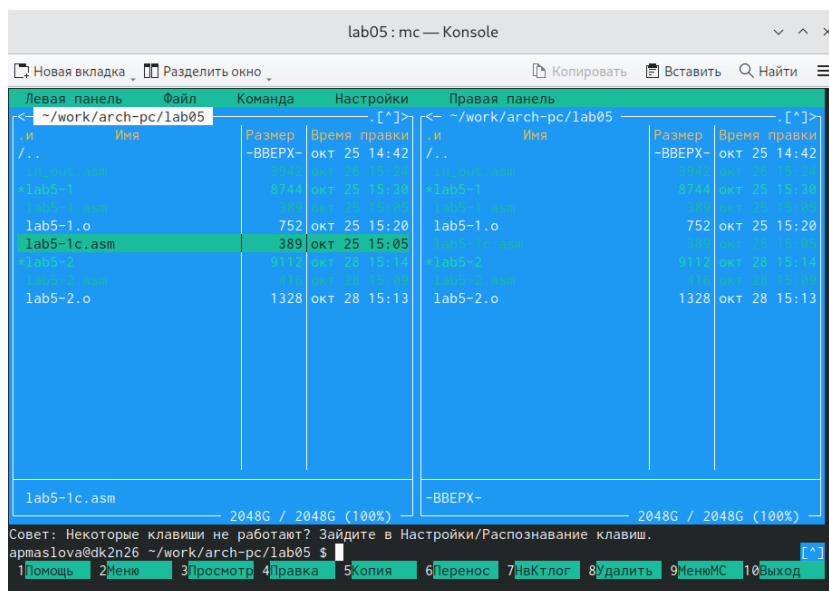


Рис. 3.1: Созданный файл *lab5-1c.asm*

Файл создан. Откроем его в редакторе *mcedit* с помощью клавиши F4.

Изменим текст программы, не используя при этом внешний файл *in_out.asm*. Нам нужно, чтобы теперь после ввода строки с клавиатуры, программа выводила введённую строку на экран. Для этого используем системный вызов *write* (в регистр *ecx* поместим значение переменной, куда записали введённую строку),

как указано в листинге 3.1 и на рис. 3.2 .

Листинг 3.1. Программа вывода сообщения “Введите строку:” на экран, ввода строки с клавиатуры и вывода введенной строки на экран

```
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
```

```
SECTION .bss
buf1: RESB 80
```

```
SECTION .text
GLOBAL _start
_start:
```

```
;Системный вызов write
```

```
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
```

```
; Системный вызов read
```

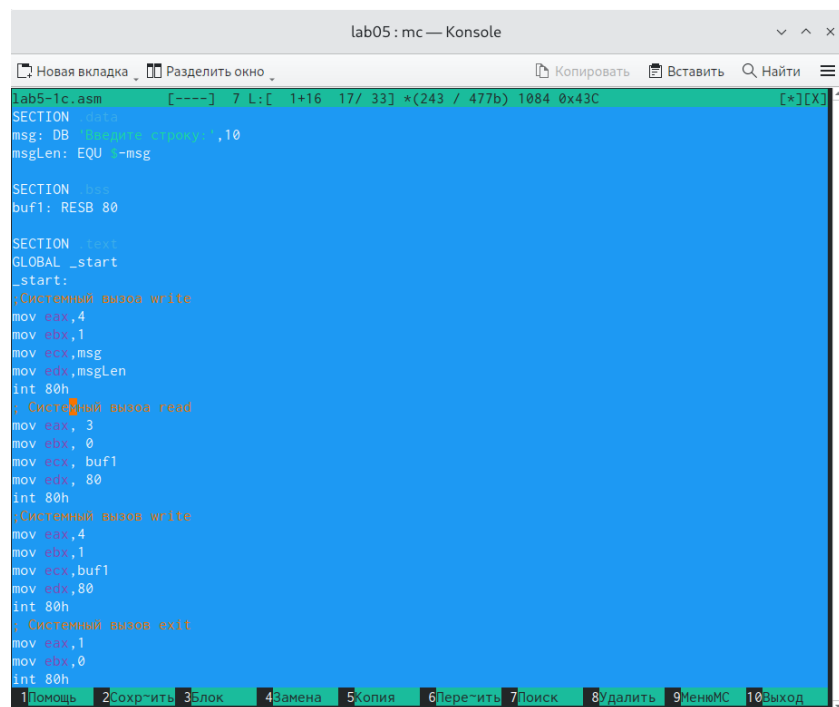
```
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h
```

```
;Системный вызов write
```

```
mov eax,4  
mov ebx,1  
mov ecx,buf1  
mov edx,80  
int 80h
```

```
; Системный вызов exit
```

```
mov eax,1  
mov ebx,0  
int 80h
```



```
lab5-1c.asm [----] 7 L: [ 1+16 17/ 33] *(243 / 477b) 1084 0x43C [*][X]  
SECTION .text  
msg: DB "Введите строку",10  
msgLen: EQU $-msg  
  
SECTION .bss  
buf1: RESB 80  
  
SECTION .text  
GLOBAL _start  
_start:  
;Системный вызов write  
mov eax,4  
mov ebx,1  
mov ecx,msg  
mov edx,msgLen  
int 80h  
; Системный вызов read  
mov eax,3  
mov ebx,0  
mov ecx,buf1  
mov edx,80  
int 80h  
;Системный вызов write  
mov eax,4  
mov ebx,1  
mov ecx,buf1  
mov edx,80  
int 80h  
; Системный вызов exit  
mov eax,1  
mov ebx,0  
int 80h  
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Переместить 7Поиск 8Удалить 9МенюМС 10Выход
```

Рис. 3.2: Текст программы в файле lab5-1c.asm

Создадим исполняемый файл и проверим его работу (рис. 3.3).

```
армаслова@dk2n26 ~ $ mc
армаслова@dk2n26 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1c.asm
армаслова@dk2n26 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1c lab5-1c.o
армаслова@dk2n26 ~/work/arch-pc/lab05 $ ./lab5-1c
Введите строку:
Маслова Анна
Маслова Анна
армаслова@dk2n26 ~ $
```

Рис. 3.3: Запуск lab5-1c

Как мы видим, программа работает корректно: на экран выводится сообщение “Введите строку:”, мы вводим строку, и эта строка затем выводится на экран.

Теперь создадим копию файла *lab5-2.asm* и зададим имя новому файлу “*lab5-2c.asm*”. Для это используем клавишу F5 (рис. 3.4).

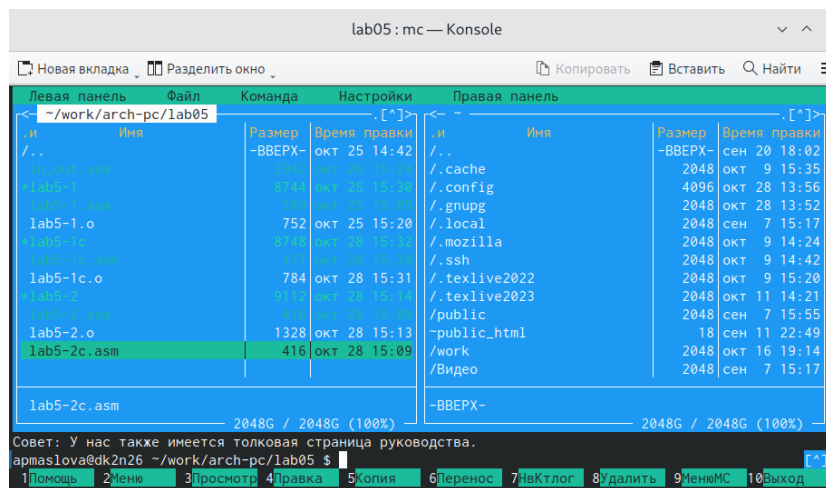


Рис. 3.4: Созданный файл lab5-2c.asm

Файл создан. Откроем его в редакторе mcedit с помощью клавиши F4.

Изменим текст программы, используя внешний файл *in_out.asm*. Нам нужно, чтобы программа работала так же, как *lab5-1c*. Для этого используем подпрограмму печати сообщения *sprintLF* (в регистр *ecx* поместим значение переменной, куда записали введенную строку), как указано в листинге 3.2 и на рис. 3.5 .

Листинг 3.2. Программа вывода сообщения “Введите строку:” на экран, ввода строки с клавиатуры и вывода введённой строки на экран с использованием внешнего файла in_out.asm

```
%include 'in_out.asm' ;подключили внешний файл

SECTION .data
msg: DB 'Введите строку:',0h
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

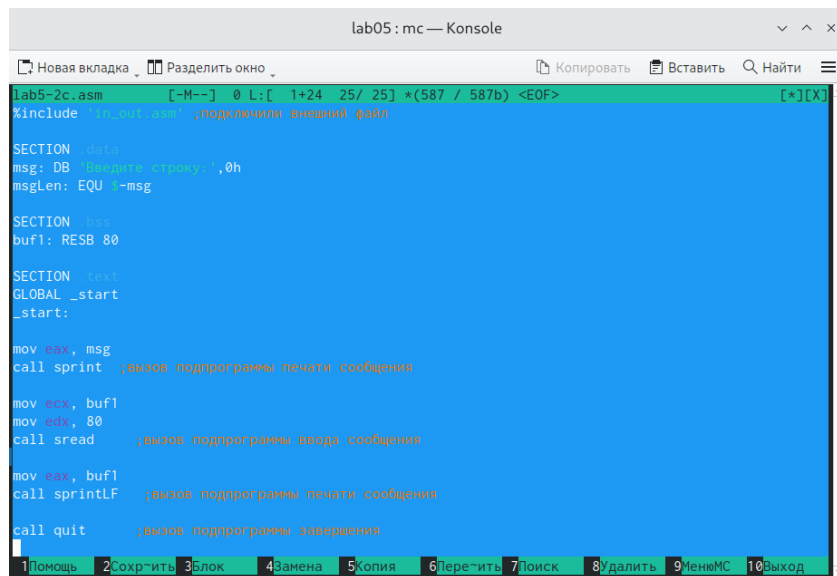
SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprint ;вызов подпрограммы печати сообщения

mov ecx, buf1
mov edx, 80
call sread ;вызов подпрограммы ввода сообщения

mov eax, buf1
call sprintLF ;вызов подпрограммы печати сообщения

call quit ;вызов подпрограммы завершения
```



```
lab5-2c.asm [-M--] 0 L:[ 1+24 25/ 25] *(587 / 587b) <EOF> [*][X]
#include "lab5-2c.h" ;подключили внешний файл

SECTION .data
msg: DB "Введите строку:",0h
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprint ;вызов подпрограммы печати сообщения

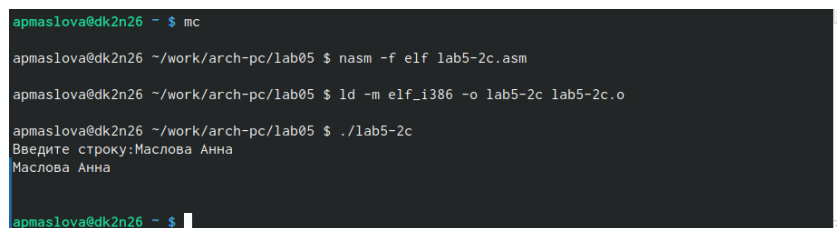
mov ecx, buf1
mov edx, 80
call sread ;вызов подпрограммы ввода сообщения

mov eax, buf1
call sprintf ;вызов подпрограммы печати сообщения

call quit ;вызов подпрограммы завершения
```

Рис. 3.5: Текст программы в файле lab5-2c.asm

Создадим исполняемый файл и проверим его работу (рис. 3.6).



```
армаслова@dk2n26 ~ $ mc
армаслова@dk2n26 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2c.asm
армаслова@dk2n26 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2c lab5-2c.o
армаслова@dk2n26 ~/work/arch-pc/lab05 $ ./lab5-2c
Введите строку:Маслова Анна
Маслова Анна
армаслова@dk2n26 ~ $
```

Рис. 3.6: Запуск lab5-2c

Как мы видим, программа работает корректно.

4 Выводы

Мы научились работать в *Midnight Commander* на практике. Освоили инструкции языка ассемблера *mov* и *int*.

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,
- 11.