

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: Архитектура компьютера

**Основы работы с Midnight Commander (mc). Структура программы
на языке ассемблера NASM. Системные вызовы в ОС GNU Linux**

Студент: ТУЙИШИМЕ Тьерри

Группа: НКАбд-05-25

Оглавление

1. Введение.....	3
2. Задание работы.....	3
3. Теоретическая часть.....	3
3,1, Структура программы на NASM.....	4
3.2. Системные вызовы.....	4
4. Выполнение работы.....	4
4.1 Основы работы с Midnight Commander.....	4
4.2 Подключение внешнего файла.....	5
4.3 Выполнение заданий для самостоятельной работы.....	6
5. Выводы.....	7
6. Список литературы.....	7

1. Введение

Данная лабораторная работа знакомит с основами работы в операционной системе GNU/Linux и предназначена для приобретения практических навыков использования:

- ✓ Файлового менеджера Midnight Commander (mc)
- ✓ Инструкций языка ассемблера mov и int
- ✓ Системных вызовов ядра Linux
- ✓ Структуры программ на ассемблере NASM
- ✓ Процесса трансляции и компоновки программ

Работа позволяет освоить основные принципы написания, отладки и выполнения программ на языке ассемблера.

2. Задание работы

1. Изучить основы работы с файловым менеджером Midnight Commander.
2. Освоить структуру программы на языке ассемблера NASM.
3. Научиться подключать внешние файлы с подпрограммами.
4. Выполнить задания для самостоятельной работы:
 - Модифицировать программу для вывода вводимой пользователем строки.
 - Сравнить работу различных подпрограмм ввода-вывода.

3. Теоретическая часть

Midnight Commander (mc)

Midnight Commander: это визуальный файловый менеджер для операционных систем Unix-like. Он предоставляет удобный текстовый интерфейс для навигации по файловой системе, работы с файлами и каталогами.

3.1, Структура программы на NASM

Программа на ассемблере NASM typically состоит из трёх секций:

- SECTION .data - секция инициализированных данных
- SECTION .bss - секция неинициализированных данных
- SECTION .text - секция кода программы

3.2. Системные вызовы

Для взаимодействия с ядром ОС используются системные вызовы через инструкцию `int 80h`.
Основные вызовы:

- ❖ `sys_write (4)` - вывод данных
- ❖ `sys_read (3)` - ввод данных
- ❖ `sys_exit (1)` - завершение программы

4. Выполнение работы

4.1 Основы работы с Midnight Commander

Шаг 1: открываем Midnight Commander, выполнив в терминале команду:

Left	File	Command	Options	Right
~				~
.n	Name	Size	Modify time	.n
..		UP--DIR	Sep 20 21:43	..
/.cache		4096	Nov 4 16:24	/.cache
/.config		4096	Nov 4 16:24	/.config
/.gnupg		4096	Nov 4 16:45	/.gnupg
/.local		4096	Sep 18 19:48	/.local
/.ssh		4096	Oct 23 21:55	/.ssh
/.stack		4096	Oct 7 11:11	/.stack
/Desktop		4096	Sep 18 19:48	/Desktop
/Documents		4096	Sep 20 13:50	/Documents
/Downloads		4096	Oct 26 15:24	/Downloads
/Music		4096	Sep 18 19:48	/Music
/Pictures		4096	Sep 19 22:28	/Pictures
/Public		4096	Sep 18 19:48	/Public
/Templates		4096	Sep 18 19:48	/Templates
/Videos		4096	Sep 18 19:48	/Videos
/demo		4096	Sep 26 21:28	/demo
/parentdir		4096	Sep 21 11:09	/parentdir
/parentdir1		4096	Sep 21 11:29	/parentdir1
/parentdir2		4096	Sep 21 11:29	/parentdir2
/parentdir3		4096	Sep 21 11:42	/parentdir3
/snap		4096	Sep 28 00:28	/snap
/study_2023-2024_arch-pc		4096	Oct 23 21:00	/study_2023-2024_arch-pc
/work		4096	Oct 23 16:26	/work
UP--DIR				UP--DIR

Шаг 2: переходим в каталог ~/work/study/2023-2024/"Архитектура Компьютера"/arch-pc

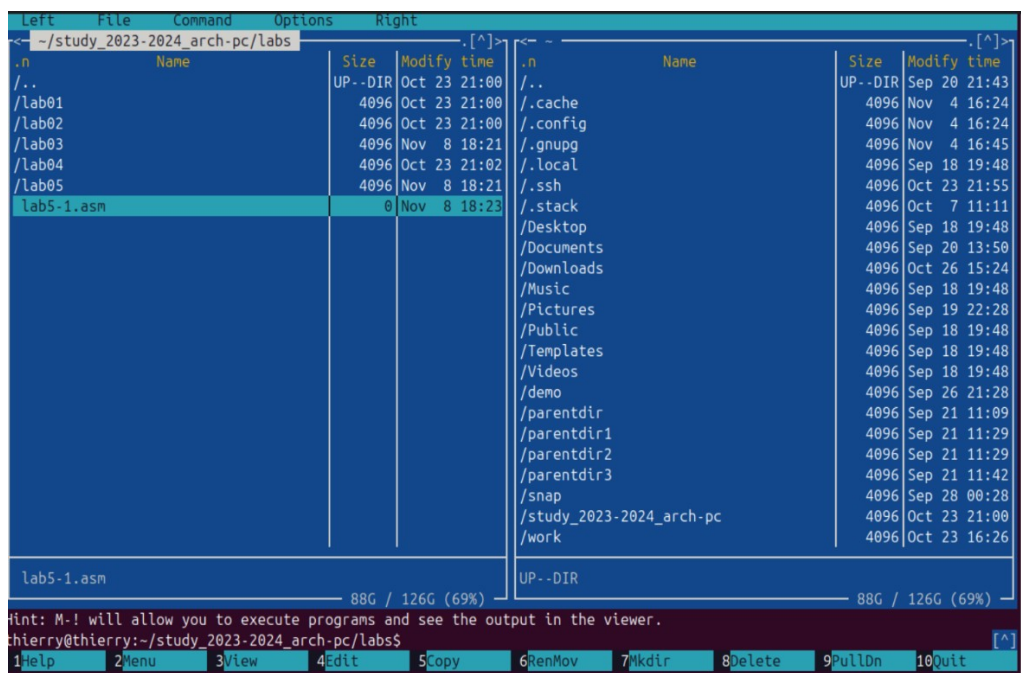
Left	File	Command	Options	Right
~/study_2023-2024_arch-pc/labs				~/study_2023-2024_arch-pc/labs
.n	Name	Size	Modify time	.n
..		UP--DIR	Oct 23 21:00	..
/lab01		4096	Oct 23 21:00	/.cache
/lab02		4096	Oct 23 21:00	/.config
/lab03		4096	Nov 8 18:21	/.gnupg
/lab04		4096	Oct 23 21:02	/.local
/lab05		4096	Nov 8 18:21	/.ssh
				/.stack
				/Desktop
				/Documents
				/Downloads
				/Music
				/Pictures
				/Public
				/Templates
				/Videos
				/demo
				/parentdir
				/parentdir1
				/parentdir2
				/parentdir3
				/snap
				/study_2023-2024_arch-pc
				/work
UP--DIR				UP--DIR

88G / 126G (70%)

Hint: M-! will allow you to execute programs and see the output in the viewer.
thierry@thierry:~/study_2023-2024_arch-pc/labs\$

1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit

Шаг 3: создаём каталог lab05 с помощью клавиши F7



Шаг 4: создаём файл `lab5-1.asm` командой:


```
thierry@thierry:~/study_2023-2024_arch-pc/labs/lab05$ nasm -f elf lab5-1.asm
thierry@thierry:~/study_2023-2024_arch-pc/labs/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
```

Шаг 7: компилируем и компоуем программу:

```
thierry@thierry:~/study_2023-2024_arch-pc/labs/lab05$ nasm -f elf lab5-1.asm
thierry@thierry:~/study_2023-2024_arch-pc/labs/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
```

Шаг 8: запускаем исполняемый файл:

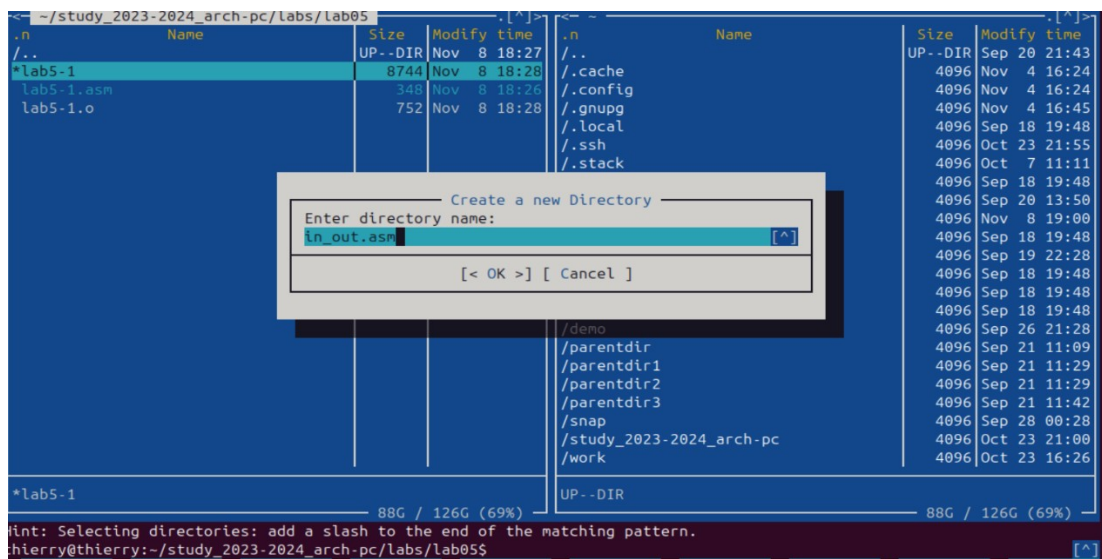
```
thierry@thierry:~/study_2023-2024_arch-pc/labs/lab05$ ./lab5-1
Введите строку:
Туйишме Тьерри
```

4.2 Подключение внешнего файла

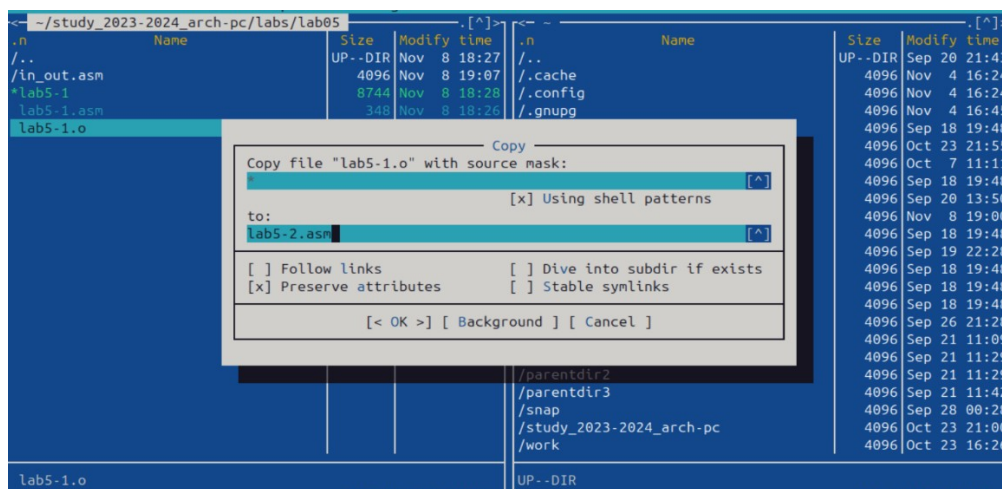
Шаг 9: скачиваем файл in_out.asm со страницы курса

Left	File	Command	Options	Right
<-	~/Downloads			.[^]>
	.n	Name	Size	Modify time
	/..		UP--DIR	Nov 4 16:46
	/Telegram Desktop		4096	Oct 23 19:05
	LAB4.pdf		1223688	Oct 23 15:33
	REPORT.docx		1438855	Sep 27 23:58
	in_out.asm		3942	Nov 8 18:55
				/.n
				/..
				/.cache
				/.config
				/.gnupg
				/.local

Шаг 10: копируем файл in_out.asm в каталог lab05 клавишей F5



Шаг 11: создаём копию файла lab5-1.asm с -2.asm



Шаг 12: Редактируем lab5-2.asm:

```

/home/thierry/study_2023-2024_arch-pc/labs/lab05/
lab5-1.asm
SECTION .data
    msg: DB 'Enter a string:',0

SECTION .bss
    buf1: RESB 80

SECTION .text
    GLOBAL _start
_start:
    mov eax, msg
    call sprintf

    mov ecx, buf1
    mov edx, 80
    call sread

    call quit

```

Шаг 13: компилируем и запускаем новую программу:

```

thierry@thierry:~/study_2023-2024_arch-pc/labs/lab05$ nasm -f elf lab5-2.asm
thierry@thierry:~/study_2023-2024_arch-pc/labs/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
thierry@thierry:~/study_2023-2024_arch-pc/labs/lab05$ ./lab5-2
Enter a string:
Tuyishime Thierry
thierry@thierry:~/study_2023-2024_arch-pc/labs/lab05$

```

Шаг 14: изменяем sprintf на sprintf в файле lab5-2.asm

```

#include 'in_out.asm'
SECTION .data
    msg: DB 'Enter a string:',0

SECTION .bss
    buf1: RESB 80

SECTION .text
    GLOBAL _start
_start:
    mov eax, msg
    call sprint

    mov ecx, buf1
    mov edx, 80
    call sread

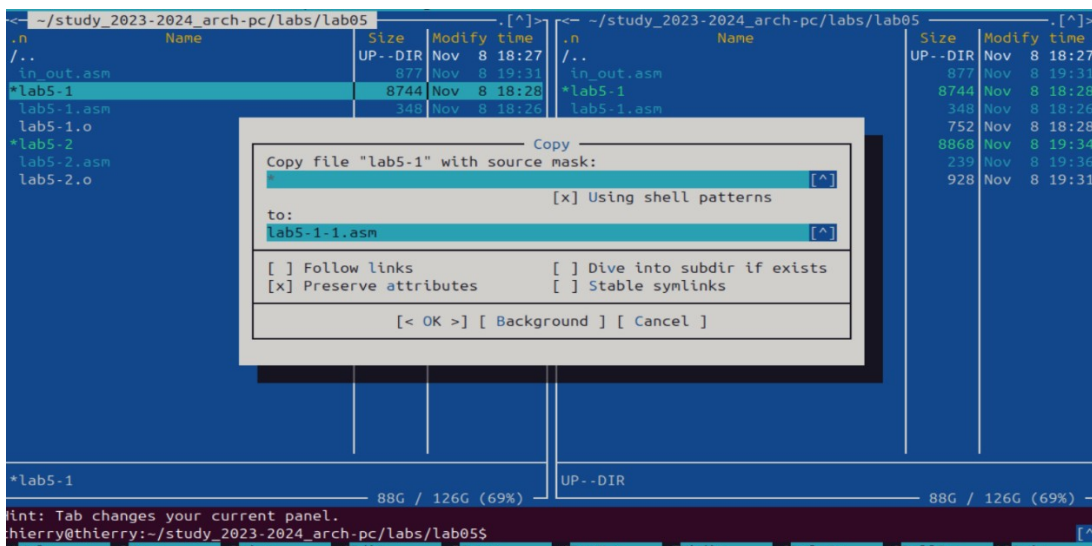
    call quit

```

Шаг 15: повторно компилируем и запускаем программу

4.3 Выполнение заданий для самостоятельной работы

Задание 1: создаём копию lab5-1.asm с именем lab5-1-1.asm



Задание 2: редактируем lab5-1-1.asm, добавляя вывод введенной строки:

Open ▾ ↗

lab5-1.asm | in_out.asm | lab5-2.asm | lab5-1-1.asm

lab5-1-1.asm

```
SECTION .data
    msg: DB 'Enter a string:',10
    msgLen: EQU $-msg

SECTION .bss
    buf1: RESB 80

SECTION .text
    GLOBAL _start
_start:
    ; Print prompt
    mov eax,4
    mov ebx,1
    mov ecx,msg
    mov edx,msgLen
    int 80h

    ; Read string
    mov eax, 3
    mov ebx, 0
    mov ecx, buf1
    mov edx, 80
```

Шаг 16: компилируем и запускаем программу:



```
lab5-1-1.asm
~/study_2023-2024_arch-pc/labs/lab05

lab5-1.asm | in_out.asm | lab5-2.asm | lab5-1-1.asm

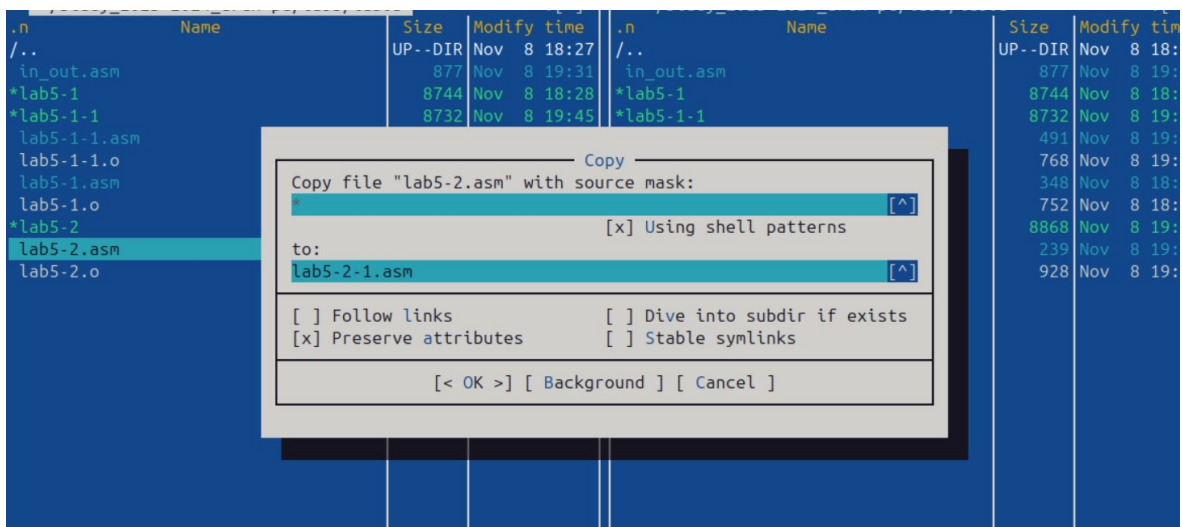
SECTION .data
    msg: DB 'Enter a string:',10
    msglen: EQU $-msg

SECTION .bss
    buf1: RESB 80

SECTION .text
    GLOBAL _start
_start:
    ; Print prompt
    mov eax,4
    mov ebx,1
    mov ecx,msg
    mov edx,msglen
    int 80h

    ; Read string
    mov eax,3
    mov ebx,0
    mov ecx,buf1
    mov edx,80
```

Задание 3: создаём копию lab5-2.asm с именем lab5-2-1.asm



Задание 4: редактируем lab5-2-1.asm для вывода введенной строки:

```
lab5-2-1.asm
~/study_2023-2024_arch-pc/labs/lab05

#include 'in_out.asm'
SECTION .data
    msg: DB 'Enter a string:',0

SECTION .bss
    buf1: RESB 80

SECTION .text
    GLOBAL _start
_start:
    mov eax, msg
    call sprint

    mov ecx, buf1
    mov edx, 80
    call sread

    ; Print entered string
    mov edx, eax    ; String length from sread
    mov eax, 4      ; sys_write
    mov ebx, 1      ; stdout
    mov ecx, buf1   ; buffer
    int 0x80
```

Шаг 17: компилируем и запускаем программу:

```
thierry@thierry:~/study_2023-2024_arch-pc/labs/lab05$ nasm -f elf lab5-2-1.asm
thierry@thierry:~/study_2023-2024_arch-pc/labs/lab05$ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
thierry@thierry:~/study_2023-2024_arch-pc/labs/lab05$ ./lab5-2-1
Enter a string:Tuyishime Thierry
Tuyishime Thierry
thierry@thierry:~/study_2023-2024_arch-pc/labs/lab05$
```

5. Выводы

В ходе выполнения лабораторной работы:

- ✓ Приобретены практические навыки работы с файловым менеджером Midnight Commander
- ✓ Освоены инструкции языка ассемблера mov и int
- ✓ Изучена структура программы на ассемблере NASM
- ✓ Освоено подключение внешних файлов с подпрограммами
- ✓ Получены навыки компиляции и компоновки программ на ассемблере
- ✓ Изучены системные вызовы для ввода-вывода данных в ОС Linux

Работа показала практическое применение языка ассемблера для взаимодействия с операционной системой на низком уровне.

6. Список литературы

1. Лабораторная работа №5 - Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM. Системные вызовы в ОС GNU Linux
2. NASM Documentation