

**Лабораторная работа No5. Основы
работы с Midnight Commander (mc).
Структура программы на языке
ассемблера NASM. Системные вызовы в
ОС GNU Linux**

Дисциплина : архитектура компьютера

Сущенко Алина Николаевна

Содержание

1	Выполнение лабораторной работы	5
2	1. Откройте МС.	6
3	2. Подключение внешнего файла in_out.asm	11
4	Выводы	19

Список иллюстраций

2.1	Рис.1 Открытие МС	6
2.2	Рис.2 Открытый МС	6
2.3	Рис.3 Переход в нужный каталог	6
2.4	Рис.4 Создание папки	7
2.5	Рис.5 Проверка наличия папки	7
2.6	Рис.6 Проверка наличия папки	7
2.7	Рис.7 Проверка наличия папки(2)	7
2.8	Рис.8 Редактор и сам текст программы	8
2.9	Рис.9 Текст программы	9
2.10	Рис.10 Транслирование	9
2.11	Рис.11 Первая проверка	9
2.12	Рис.12 Вторая проверка после компиляции	9
2.13	Рис.13 Запуск файла	10
2.14	Рис.14 Запуск файла	10
3.1	Рис.15 Две открытые панели и скачанный файл	11
3.2	Рис.16 Скопированный файл	11
3.3	Рис.17 Создание копии файла	12
3.4	Рис.18 Проверка наличия файла	12
3.5	Рис.19 Сам текст файла	13
3.6	Рис.20 Проверка наличия файлов	13
3.7	Рис.21 Запуск файла	14
3.8	Рис.22	14
3.9	Рис.23 Проверка работы	14
3.10	Рис.24 Создание копии файла	15
3.11	Рис.24-1 Текст программы	16
3.12	Рис.26 Проверка	17
3.13	Рис.27 Созданная копия файла	17
3.14	Рис.27-1 Текст программы	18
3.15	Рис.28 Текст программы	18

Список таблиц

#Задания 1. Откройте Midnight Commander 2. Создать папку lab05 3. Создать файл “lab5-1.asm”, оттранслировать текст программы, скомпонировать объектный файл, после чего запустить полученный файл. 4. Сказать файл с ТУИС и скопировать в каталог lab05. 5. Скопировать файлы “lab5-1.asm”, переименовать в “lab5-2.asm” и переделать текст программы так, чтобы использовалась программа из скаченного файла с ТУИС. 6. Создать файл и проверить его работу. 7. Создать копию файла “lab5-1.asm”. Изменить текст программы так, чтобы не был использован внешний файл “in_out.asm”, но при этом сохранять последовательность действий по алгоритму. 8. Создать копию “lab5-2.asm”, изменить текст программы(с использованием файла in_out.asm), так же припрдерживаясь алгоритма. 9. Создать файлы соответствующие условию заданий и проверить из работу. # Цель работы

Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int.

1 Выполнение лабораторной работы

2 1. Откройте МС.

Открываем midnight commander с помощью команды 'mc' (рис.1 [fig:001 width=70%]) ,

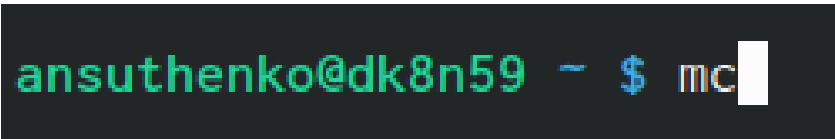


Рис. 2.1: Рис.1 Открытие МС

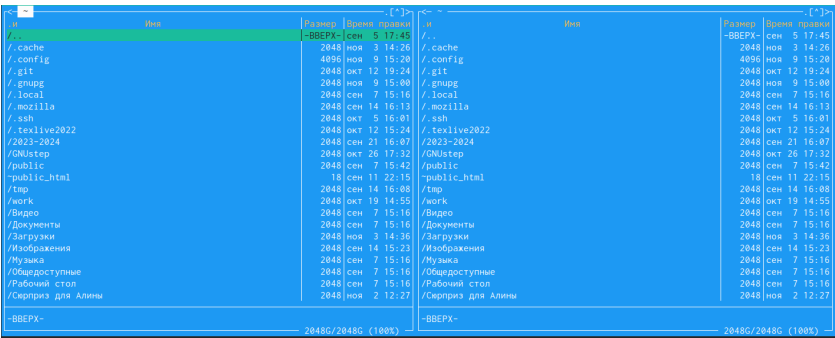


Рис. 2.2: Рис.2 Открытый МС

2. Используя клавиши на клавиатуре переходим в нужный нам каталог и нажимаем 'enter' (рис.3 [fig:003 width=70%]) .

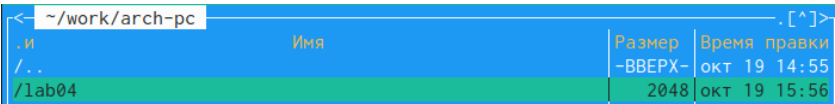


Рис. 2.3: Рис.3 Переход в нужный каталог

4. С помощью клавиши 'F7' создаем папку 'lab05' и переходим в неё (рис.4 [fig:004 width=70%]) , (рис.5 [fig:005 width=70%])

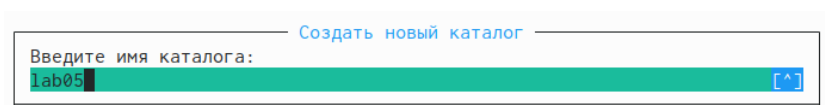


Рис. 2.4: Рис.4 Создание папки

Имя	Размер	Время правки
./	-ВВЕРХ-	окт 19 14:55
./lab04	2048	окт 19 15:56
./lab05	2048	ноя 9 15:08

Рис. 2.5: Рис.5 Проверка наличия папки

5. С помощью команды 'touch' создаём файл 'lab5-1.asm' и проверяем его наличие (рис.6 [fig:006 width=70%]) , (рис.7 [fig:007 width=70%])

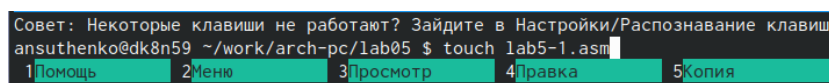


Рис. 2.6: Рис.6 Проверка наличия папки

Имя	Размер	Время правки
./	-ВВЕРХ-	ноя 9 15:23
lab5-1.asm	0	ноя 9 15:09

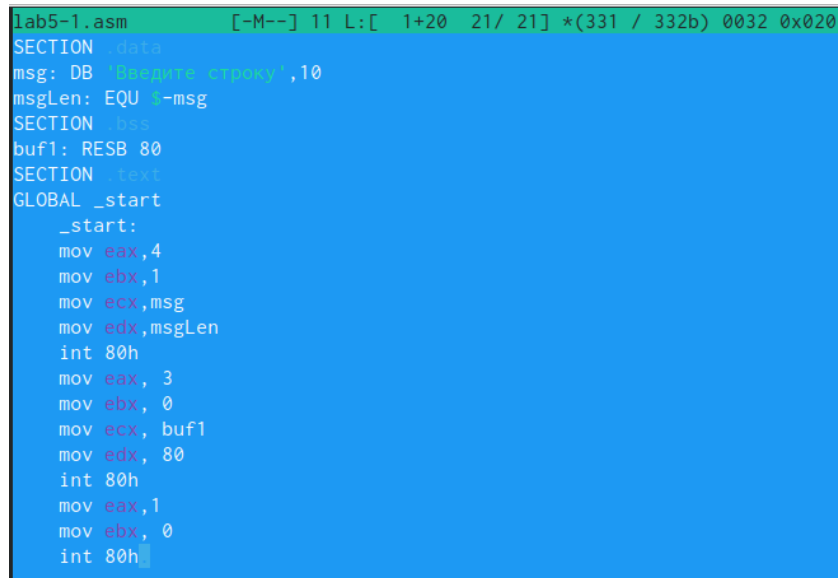
Рис. 2.7: Рис.7 Проверка наличия папки(2)

6. С помощью клавиши 'F4' переходим в редактирование файла и вставляем текст программы предложенном на ТУИС.

Текст программы имеет вид: SECTION .data msg: DB 'Введите строку:',10 msgLen:
EQU \$-msg SECTION .bss 13 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,1 mov ebx,0 int 80h

(рис.8 [fig:008 width=70%])



```
lab5-1.asm      [-M--] 11 L:[ 1+20 21/ 21] *(331 / 332b) 0032 0x020
SECTION .data
msg: DB 'Введите строку',10
msgLen: EQU 1-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,1
    mov ebx, 0
    int 80h
```

Рис. 2.8: Рис.8 Редактор и сам текст программы

7. С помощью клавиши 'F2' сохраняем проделанные изменения и просматриваем результат с помощью клавиши 'F3' (рис.9 [fig:009 width=70%])


```

/afs/.dk.sci.pfu.edu.ru/home/a/n/ansuthenko/work/arch-pc/lab05/lab5-1.asm
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,1
    mov ebx,0
    int 80h

```

Рис. 2.9: Рис.9 Текст программы

8. Транслируем текст программы в объектный файл, компилируем начальный файл и передаём объектный файл на обработку компоновщику (рис.10 [fig:010 width=70%]), (рис.11 [fig:011 width=70%]), (рис.12 [fig:012 width=70%])

```

ansuthenko@dk8n59 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm

```

Рис. 2.10: Рис.10 Транслирование

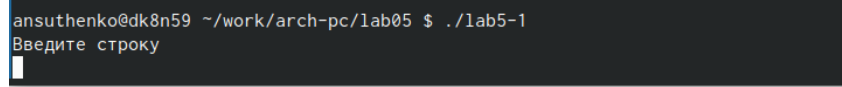
~/work/arch-pc/lab05		. [^]>	
Имя	Размер	Время правки	
..	-ВВЕРХ-	ноя 9 15:23	
lab5-1.asm	332	ноя 9 15:19	
lab5-1.o	752	ноя 9 15:21	

Рис. 2.11: Рис.11 Первая проверка

~/work/arch-pc/lab05		. [^]>	
Имя	Размер	Время правки	
..	-ВВЕРХ-	ноя 9 15:23	
lab5-1	8748	ноя 9 15:26	
lab5-1.asm	332	ноя 9 15:19	
lab5-1.o	752	ноя 9 15:27	

Рис. 2.12: Рис.12 Вторая проверка после компиляции

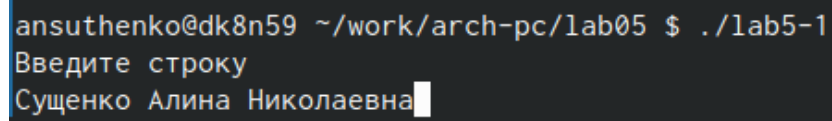
9. Пробуем запустить созданный файл. (рис.13 [fig:013 width=70%])



```
ansuthenko@dk8n59 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку
█
```

Рис. 2.13: Рис.13 Запуск файла

10. Ввод фамилии в строке. (рис.14 [fig:014 width=70%])



```
ansuthenko@dk8n59 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку
Сущенко Алина Николаевна█
```

Рис. 2.14: Рис.14 Запуск файла

3 2. Подключение внешнего файла in_out.asm

1. Скачиваем файл in_out.asm с ТУИС и для удобства открываем его в соседнем каталоге (рис.15 [fig:015 width=70%])

Имя	Размер	Время	Правки
in_out.asm.save	3944	ноя 3 14:31	

Рис. 3.1: Рис.15 Две открытые панели и скачанный файл

2. Используя клавишу 'F5' копируем файл in_out.asm в тот же каталог с нашим файлом (рис.16 [fig:016 width=70%])

Имя	Размер	Время	Правки
in_out.asm.save	3944	ноя 3 14:31	
lab5-1	8748	ноя 9 15:38	
lab5-1.asm	332	ноя 9 15:19	
lab5-1.o	752	ноя 9 15:37	

Рис. 3.2: Рис.16 Скопированный файл

3. С помощью той же клавиши создаём копию файла 'lab5-1.asm' с именем 'lab5-2.asm'

Текст программы имеет вид: %include 'in_out.asm' SECTION .data msg: DB 'Введите строку:',10 SECTION .bss buf1: RESB 80 SECTION .text GLOBAL _start 22 _start: mov eax, msg call sprintLF mov ecx, buf1 mov edx, 80 call sread call quit

(рис.17 [fig:017 width=70%]) , (рис.18 [fig:018 width=70%]) , (рис.19 [fig:019 width=70%])

Копирование

Копировать файл "lab5-1.asm" с исходным шаблоном:

[^]

[x] Метасимволы shell

В:

[^]

[] Разыменовывать ссылки [] Внутрь подкаталога, если есть
[x] Сохранять атрибуты [] Изменять относительные ссылки

[< Дальше >] [В фоне] [Прервать]

Рис. 3.3: Рис.17 Создание копии файла

in_out.asm.save	3944	ноя 3 14:31
lab5-1	8715	ноя 8 15:38
lab5-1.asm	332	ноя 9 15:19
lab5-1.o	752	ноя 9 15:37
lab5-2.asm	332	ноя 9 15:19

Рис. 3.4: Рис.18 Проверка наличия файла

12

```

#include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку',0h
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

```

Рис. 3.5: Рис.19 Сам текст файла

4. Транслируем текст программы (скрины не сохранились) в объектный файл и выполняем компоновку объектного файла, после чего производим запуск программы. (рис.20 [fig:020 width=70%]) , (рис.21 [fig:021 width=70%])

```

ansuthenko@dk8n59 ~/work/arch-pc/lab05 $ ls
in_out.asm  lab5-1  lab5-1.asm  lab5-1.o  lab5-2.asm  lab5-2.o

```

Рис. 3.6: Рис.20 Проверка наличия файлов

```
ansuthenko@dk6n52 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Сущенко Алина
```

Рис. 3.7: Рис.21 Запуск файла

5. Меняем в файле 'lab5-2.asm' 'sprintLF' на 'sprint'

Текст программы выглядит так: %include 'in_out.asm' SECTION .data msg: DB 'Введите строку:',10 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 (рис.22 [fig:022 width=70%])

```
lab5-2.asm [-M--] 11 L: [ 1+13 14/ 20] *(171 / 222b) 0010 0x00A
%include "in_out.asm"

SECTION .data
msg: DB 'Введите строку:',0h

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
```

Рис. 3.8: Рис.22

6. Создаём исполняемый файл и проверяем его работу и порверяем его работу (рис.23 [fig:023 width=70%])

```
ansuthenko@dk6n52 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
ansuthenko@dk6n52 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
ansuthenko@dk6n52 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Сущенко Алина
```

Рис. 3.9: Рис.23 Проверка работы

В файле lab5-2.asm замените подпрограмму sprintLF на sprint. Создайте исполняемый файл и проверьте его работу. В чем разница?

Разница в том, что в первом случае ввод текста с клавиатуры производится на следующей строке, а во втором - сразу после двоеточия.

#Задание для самостоятельной работы

1. Создаём копию файла 'lab5-1.asm' с новым именем 'lab5-1-1.asm' (рис.24 [fig:024 width=70%])

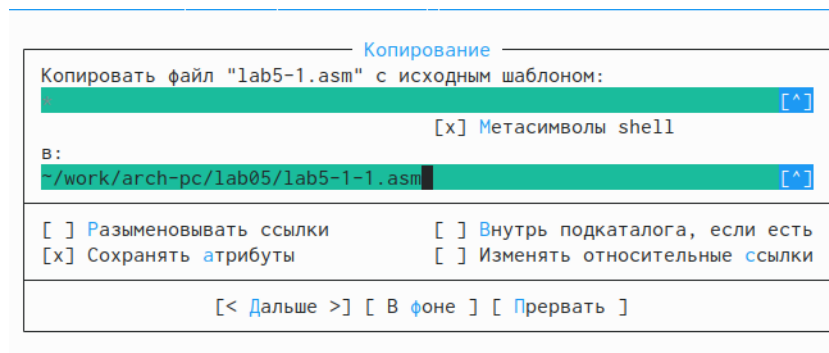


Рис. 3.10: Рис.24 Создание копии файла

2. Без использования программы in_out.asm вносим изменения в программу, чтобы она работала по заданному алгоритму: 1)Вывести приглашение типа "Введите строку";2)Вывести строку с клавиатуры;3)Вывести введенную строку на экран (рис.24-1 [fig:025 width=70%]) Текст программы: 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,1 mov ebx,0 int 80h

```

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. 3.11: Рис.24-1 Текст программы

3. Получаем исполняемый файл, проверяем работу. Вводим свою фамилию и имя (рис.26 [fig:026 width=70%])

```
ansuthenko@dk6n52 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1-1.asm
ansuthenko@dk6n52 ~/work/arch-pc/lab05 $ ls
in_out.asm lab5-1 lab5-1-1.asm lab5-1-1.o lab5-1.asm lab5-1.o lab5-2 lab5-2.asm lab5-2.o list5-1.lst main5
ansuthenko@dk6n52 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
ansuthenko@dk6n52 ~/work/arch-pc/lab05 $ ls
in_out.asm lab5-1 lab5-1-1 lab5-1-1.asm lab5-1-1.o lab5-1.asm lab5-1.o lab5-2 lab5-2.asm lab5-2.o list5-1.lst main5
ansuthenko@dk6n52 ~/work/arch-pc/lab05 $ ./lab5-1-1
Введите строку
Суценко Алина
Суценко Алина
```

Рис. 3.12: Рис.26 Проверка

4. Создаём копию файла 'lab5-2.asm' и переименовываем копию 'lab5-2-2.asm' (рис.27 [fig:027 width=70%])

```
lab5-1.o
lab5-1
lab5-2-2.asm
```

Рис. 3.13: Рис.27 Созданная копия файла

5. Вносим изменения в изначальный код, чтобы она работала по алгоритму:1) Вывести приглашение “Ввести строку:”;2)Вывести строку с клавиатуры;3)Вывести введённую строку на экран

Текст программы: %include 'in_out.asm' SECTION .data msg: DB 'Введите строку:',10 SECTION .bss buf1: RESB 80 SECTION .text GLOBAL _start _start: mov eax, msg call sprint mov ecx, buf1 mov edx, 80 call sread mov eax,4 mov ebx,1 mov ecx,buf1 int 80h call quit

(рис.27-1 [fig:028 width=70%])

```

#include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
mov eax, 4
mov ebx, 1
mov ecx, buf1
int 80h
call quit

```

Рис. 3.14: Рис.27-1 Текст программы

6. Создаем файл и заставляем его работать. (рис.28 [fig:029 width=70%])

```

ansuthenko@dk6n52 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2-2.asm
ansuthenko@dk6n52 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2-2 lab5-2-2.o
ansuthenko@dk6n52 ~/work/arch-pc/lab05 $ ./lab5-2-2
Введите строку: Сущенко Алина
Сущенко Алина

```

Рис. 3.15: Рис.28 Текст программы

4 Выводы

В ходе выполнения этой работы мы приобрели навыки работы в МС, а так же освоили инструкции языка ассемблера `mov` и `int`.