

Лабораторная работа №5

Архитектура вычислительных систем

Сабралиева Марворид Нуралиевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	16
	Список литературы	17

Список иллюстраций

3.1	Открытие МС	7
3.2	Переход в каталог	8
3.3	Создание папки	8
3.4	Создание файла	8
3.5	Редактор mscedit	9
3.6	Проверим файл	10
3.7	Оттранслируем текст	10
3.8	Запуск файла	10
3.9	Проверка	11
3.10	Скачиваем файл	11
3.11	Копируем файл	12
3.12	Копируем файл	12
3.13	Исправление текста	13
3.14	Проверим файл	13
3.15	Замена	13
3.16	Проверим файл	14
3.17	Создадим копию	14
3.18	Проверим файл	14
3.19	Копия файла	14
3.20	Копия файла	15
3.21	Копия файла	15

Список таблиц

1 Цель работы

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

2 Задание

1. Создайте копию файла lab6-1.asm. Внесите изменения в программу (без использования внешнего файла in_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.
3. Создайте копию файла lab6-2.asm. Исправьте текст программы с использование подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.
4. Создайте исполняемый файл и проверьте его работу.

3 Выполнение лабораторной работы

1. Откроем Midnight Commander: 3.1

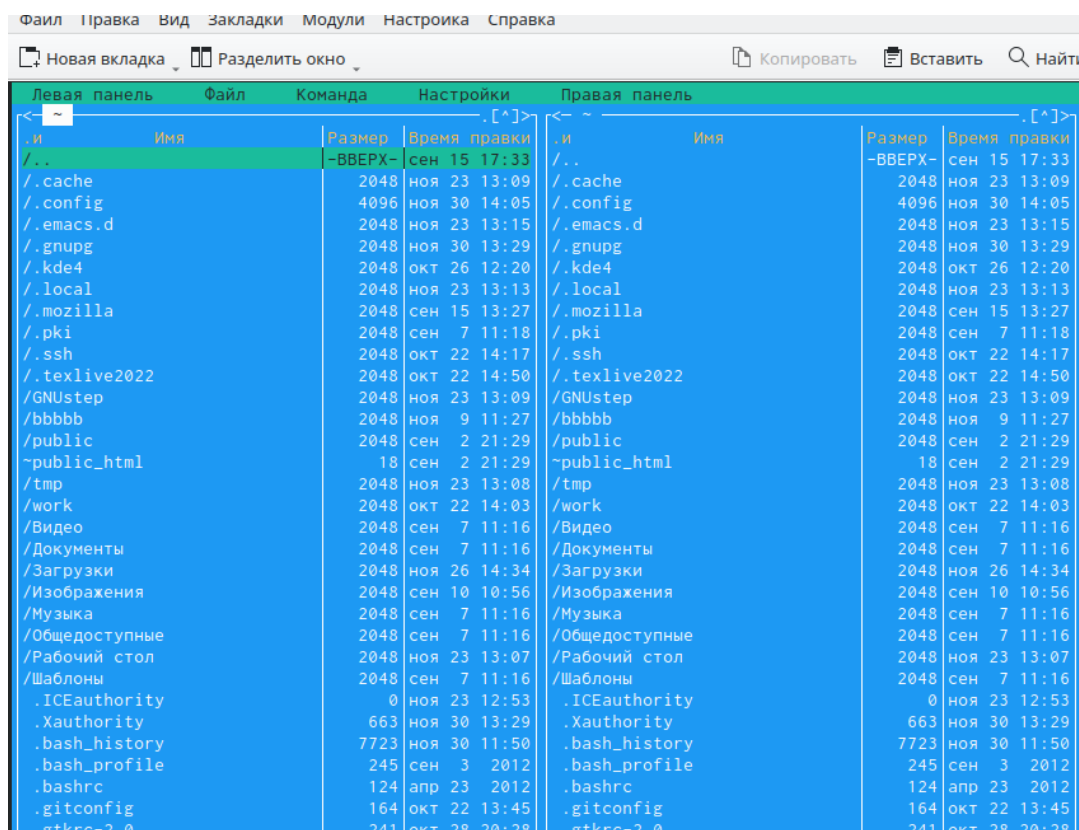


Рис. 3.1: Открытие MC

2. Пользуясь клавишами клавиатуры перейдем в каталог arch-rc 3.2

...2-2023/Архитектура компьютера/arch-pc				~			
.и	Имя	Размер	Время правки	.и	Имя	Размер	Время правки
./..		-ВВЕРХ-	окт 22 14:33	./..		-ВВЕРХ-	сен 15 17:33
./.git		2048	ноя 30 14:16	./.cache		2048	ноя 23 13:09
./config		2048	окт 22 14:34	./config		4096	дек 7 11:00
./lab04		2048	ноя 9 15:55	./emacs.d		2048	ноя 23 13:15
./labs		2048	окт 22 14:36	./gnupg		2048	дек 7 10:59

Рис. 3.2: Переход в каталог

- С помощью функциональной клавиши F7 создаем папку lab05 и переходим в созданный каталог: 3.3

Создать новый каталог

Введите имя каталога:

lab05

[< Дальше >]
[Прервать]

/Музыка

Рис. 3.3: Создание папки

- Пользуясь строкой ввода и командой touch создаем файл lab5.asm 3.4

Левая панель		Файл	Команда	Настройки
< .../Архитектура компьютера/arch-pc/lab05				~.[^]>
.и	Имя	Размер	Время правки	
./..		-ВВЕРХ-	дек 7 11:10	
lab5.asm		0	дек 7 11:13	

Рис. 3.4: Создание файла

- С помощью функциональной клавиши F4 откроем файл lab5.asm для редактирования во встроенном редакторе. В нашем случае это редактор mcedit 3.5


```

lab5.asm          [-M--] 20 L:[ 1+34 35/ 35] *(2462/2462b) <EOF>
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
                                ; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
;----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;----- Системный вызов 'write' -----
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
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
;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
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
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
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: Редактор mcedit

6. Введем в текст программы, сохраним изменения и проверим результат с помощью клавиши F3 fig. 3.6

```

/afs/.dk.sci.pfu.edu.ru/home/m/n/mn~a компьютера/arch-pc/lab05/lab5.asm 2462/2462
;
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
;----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;----- Системный вызов 'write' -----
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
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
;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
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
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

```

Рис. 3.6: Проверим файл

7. Оттранслируем текст программы lab5.asm в объектный файл. 3.7

```

mnsabralieva@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ab5.asm
1Помощь 2Меню 3Просмотр 4Правка 5Копия 6Перенос 7НаКтлог 8Удалить 9МенюМС 10Выход

```

Рис. 3.7: Оттранслируем текст

8. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введем свои ФИО 3.8

```

mnsabralieva@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 -o lab5 lab5.o
mnsabralieva@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./lab5
Введите строку:
Сабралиева Марворид Нуралиевна

```

Рис. 3.8: Запуск файла

Проверим полученные файлы 3.9

Левая панель	Файл	Команда	Настройки	Прав
<	...-2023/Архитектура компьютера/arch-pc/lab05			[.^[>
.и	Имя	Размер	Время правки	
/..		-ВВЕРХ-	дек 7 11:10	
*lab5		8744	дек 7 11:33	
lab5.asm		2462	дек 7 11:18	
lab5.o		752	дек 7 11:21	

Рис. 3.9: Проверка

9. Скачаем файл in_out.asm со страницы курса в ТУИС. 3.10

```

Открыть  in_out.asm  Сохранить
~/Загрузки

1 ;----- slen -----
2 ; Функция вычисления длины сообщения
3 slen:
4     push    ebx
5     mov     ebx, eax
6
7 nextchar:
8     cmp     byte [eax], 0
9     jz      finished
10    inc     eax
11    jmp     nextchar
12
13 finished:
14    sub     eax, ebx
15    pop     ebx
16    ret
17
18
19 ;----- sprint -----
20 ; Функция печати сообщения
21 ; входные данные: mov eax, <message>
22 sprint:
23     push    edx
24     push    ecx
25     push    ebx
26     push    eax
27     call    slen
28
29     mov     edx, eax
30     pop     eax
31
32     mov     ecx, eax
33     mov     ebx, 1
34     mov     eax, 4

```

Рис. 3.10: Скачиваем файл

10. Подключаемый файл in_out.asm должен лежать в том же каталоге, что и

файл с программой, в которой он используется. Скопируем файл in_out.asm в каталог с файлом lab5.asm 3.11

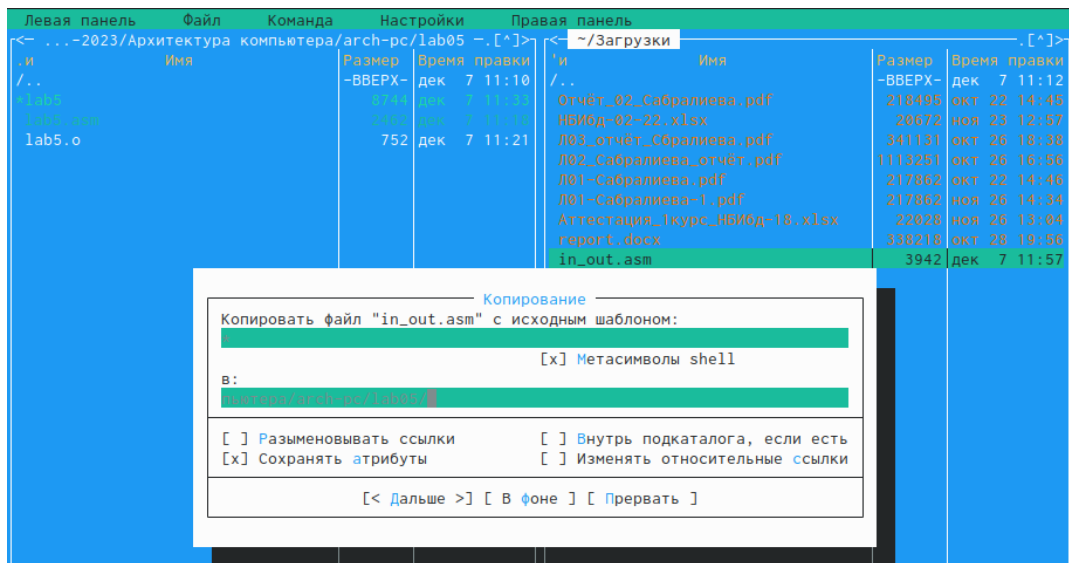


Рис. 3.11: Копируем файл

11. С помощью функциональной клавиши F6 создайте копию файла lab5.asm с именем lab5-2.asm fig. 3.12

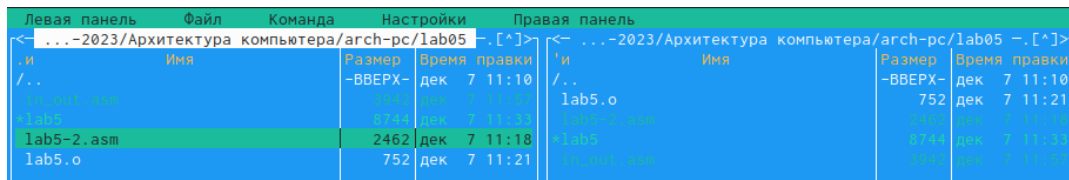


Рис. 3.12: Копируем файл

12. Исправим текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm fig. 3.13

```

;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных 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 read ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 3.13: Исправление текста

Создадим исполняемый файл и проверим его работу: fig. 3.14

```

mnsabralieva@dk1n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf lab5-2.asm
mnsabralieva@dk1n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
mnsabralieva@dk1n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./lab5-2
Введите строку: Marvoriid
mnsabralieva@dk1n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $

```

Рис. 3.14: Проверим файл

13. В файле lab6-2.asm заменим подпрограмму sprintf на printf. fig. 3.15

```

/afs/.dk.sci.pfu.edu.ru/home/m/n/mnsabralieva@dk1n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05/lab5-2.asm 1222/1222 100%
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных 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 printf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в 'EAX'
mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
call read ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 3.15: Замена

printf – работает аналогично printf, но при выводе на экран добавляет к сообщению символ перевода строки. Создадим исполняемый файл и проверим его работу: fig. 3.16

```

mnsabralieva@dk1n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf lab5-2.asm
mnsabralieva@dk1n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
mnsabralieva@dk1n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./lab5-2
Введите строку: Marvolid
Marvolid
mnsabralieva@dk1n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $

```

Рис. 3.16: Проверим файл

14. Создадим копию файла lab5.asm. Внесем изменения в программу (без использования внешнего файла in_out.asm) fig. 3.17

```

mnsabralieva@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ cp ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05/lab5.asm lab5-3.asm
mnsabralieva@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $

```

Рис. 3.17: Создадим копию

15. Проверим его работу: fig. 3.18

```

mnsabralieva@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf lab5-3.asm
mnsabralieva@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
mnsabralieva@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./lab5-3
Введите строку:
Сабралиева
Сабралиева
mnsabralieva@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $

```

Рис. 3.18: Проверим файл

16. Создадим копию файла lab6-2.asm: fig. 3.19

Копирование

Копировать файл "lab5-2.asm" с исходным шаблоном:

[^]

[x] Метасимволы shell

В:

[^]

пьютера/arch-pc/lab05/lab5-4.asm

☐ Разыменовывать ссылки
☒ Сохранять атрибуты

☐ Внутрь подкаталога, если есть
☐ Изменять относительные ссылки

[< Дальше >]
[В фоне]
[Прервать]

Рис. 3.19: Копия файла

Исправим текст программы с использованием подпрограмм из внешнего файла in_out.asm: fig. 3.20

```
-----  
; Программа вывода сообщения на экран и ввода строки с клавиатуры  
-----  
  
%include 'in_out.asm'      ; подключение внешнего файла  
  
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, buf1 ;  
call sprint ;  
call quit                  ; вызов подпрограммы завершения
```

Рис. 3.20: Копия файла

17. Создадим исполняемый файл и проверим его работу: fig. 3.21

```
mnsabralieva@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf lab5-4.asm  
mnsabralieva@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 -o lab5-4 lab5-4.o  
mnsabralieva@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./lab5-4  
Введите строку: Сабралиева  
Сабралиева  
mnsabralieva@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $
```

Рис. 3.21: Копия файла

4 Выводы

Мы приобрели практические навыки работы в Midnight Commander и освоили инструкции языка ассемблера `mov` и `int`.

Список литературы