

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: Архитектура компьютера

Мошаров Денис Максимович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Порядок выполнения лабораторной работы	7
3.2	Задание для самостоятельной работы	11
4	Выводы	12

Список иллюстраций

3.1	мс	7
3.2	Переходим в каталог	7
3.3	Создаем каталог	8
3.4	touch	8
3.5	Открываем файл, заполняем	9
3.6	Открываем файл и убеждаемся, что файл содержит текст программы	9
3.7	Проверка	10
3.8	Скачиваем файл и копируем файл в нужную директорию	10
3.9	Создаем копию файла	10
3.10	Смотрим, как работает программа и сравниваем с прошлой	11
3.11	Проверяем правильность	11
3.12	Редактируем файл	11
3.13	Проверяем правильность программы	11

Список таблиц

1 Цель работы

Освоить инструкции языка ассемблера mov. Приобрести знания использования Midnight Commander.

2 Задание

Написать 2 программы по примеру и изменить их структуру по условию.

3 Выполнение лабораторной работы

3.1 Порядок выполнения лабораторной работы

Открываем Mid. Commander

mc

Рис. 3.1: mc

Переходим в каталог, созданный при выполнении 4 ЛБ

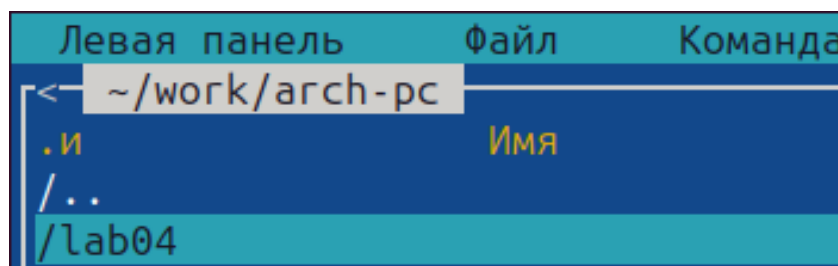


Рис. 3.2: Переходим в каталог

Создаем каталог lab05

Левая панель	Файл	Команда
< ~ /work/arch-pc/lab05		. [^]>
.и	Имя	Размер
/. ..	-BBERX-	Время правки
lab5-1.asm	0	Dec 25 20:37
		Dec 25 20:39

Рис. 3.3: Создаем каталог

Создаем файл lab5-1.asm

Левая панель	Файл	Команда
< ~ /work/arch-pc/lab05		. [^]>
.и	Имя	Размер
/. ..	-BBERX-	Время правки
lab5-1.asm	0	Dec 25 20:37
		Dec 25 20:39

Рис. 3.4: touch

Открываем файл для редактирования и заполняем его по листингу


```

GNU nano 7.2 /home/dmmosharov/work/arch-pc/lab05/lab5-1.asm
;-----
; 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
;-----

```

Рис. 3.5: Открываем файл, заполняем

Открываем файл и просматриваем

```

/home/dmmosharov/work/arch-pc/lab05/lab5-1.asm 2323/2516 9
;-----
; 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
;-----
1Помощь 2Раз-рн 3Выход 4Нех 5Пер-ти 6 7Поиск 8Исх-ый 9Формат10Выхо

```

Рис. 3.6: Открываем файл и убеждаемся, что файл содержит текст программы

Транслируем текст программы и запускаем файл

```
dmmosharov@dmmosharov:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
dmmosharov@dmmosharov:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
dmmosharov@dmmosharov:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
dmmosharov
dmmosharov@dmmosharov:~/work/arch-pc/lab05$
```

Рис. 3.7: Проверка

Скачиваем файл и копируем файл в нужную директорию

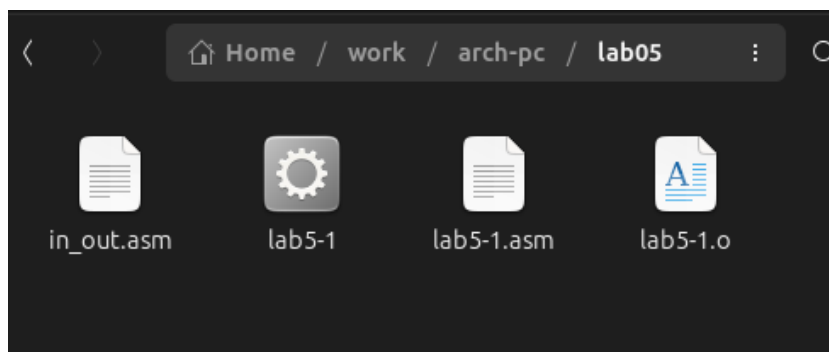


Рис. 3.8: Скачиваем файл и копируем файл в нужную директорию

Создаем копию файла

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

Рис. 3.9: Создаем копию файла

Транслируем и запускаем файл

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

Рис. 3.10: Смотрим, как работает программа и сравниваем с прошлой

3.2 Задание для самостоятельной работы

Создаем копию файла lab5-1.asm Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль Транслируем файл и запускаем программу

```
dmmosharov@dmmosharov:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
dmmosharov@dmmosharov:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
dmmosharov@dmmosharov:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
dmmosharov
dmmosharov@dmmosharov:~/work/arch-pc/lab05$
```

Рис. 3.11: Проверяем правильность

Создаем копию файла lab5-2.asm Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль

Редактируем файл

Рис. 3.12: Редактируем файл

Транслируем файл и запускаем

```
dmmosharov@dmmosharov:~/work/arch-pc/lab05$ nasm -f elf lab5-4.asm
dmmosharov@dmmosharov:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-4 lab5-4.o
dmmosharov@dmmosharov:~/work/arch-pc/lab05$ ./lab5-4
Введите строку: dmmosharov
dmmosharov
```

Рис. 3.13: Проверяем правильность программы

4 Выводы

Мы приобрели навыки работы с Midnight Commander и освоили инструкции mov.