

Отчет по лабораторной работе № 5

Архитектура вычислительных систем

Кулагина Анна Сергеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	14
	Список литературы	15

Список иллюстраций

3.1	1.png	8
3.2	4.png	11
3.3	5.png	12

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 lab5-1.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 lab5-2.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.

3 Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.1) 1)Откроем Midnight Commander,пользуясь клавишами ,  и Enter перейдем в каталог ~/work/arch- pc ,созданный при выполнении лабораторной работы No4.С помощью функциональной клавиши F7 создадим папку lab05 и перейдем в него.

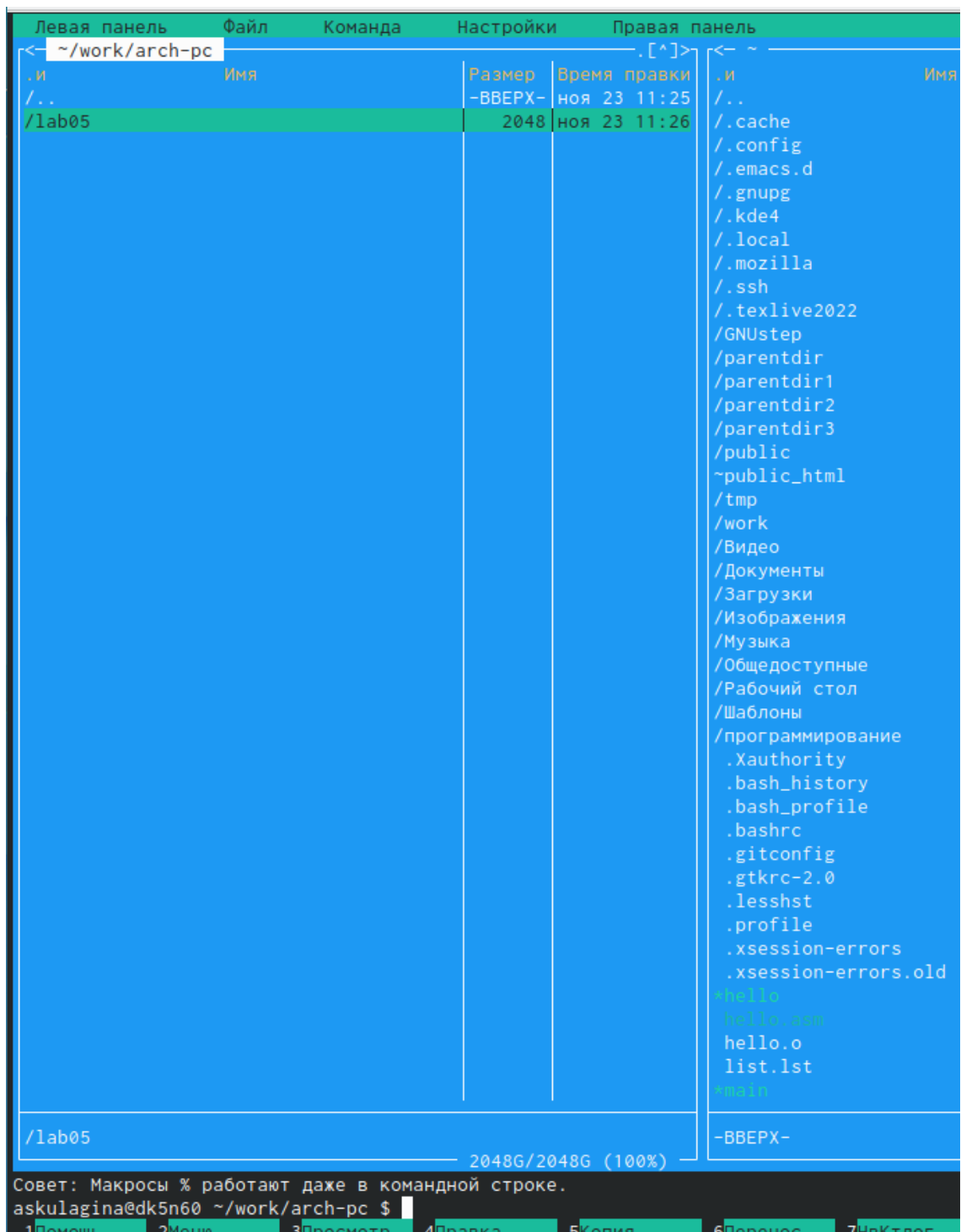


Рис. 3.1: 1.png

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. ??)

- 2) С помощью функциональной клавиши F4 откроем файл lab5-1.asm для редактирования во встроенном редакторе. В качестве встроенного редактора Midnight Commander используется редакторы nano или mcedit. Введем текст программы из листинга сохраним изменения и закроем файл. С помощью функциональной клавиши F3 откроем файл lab5-1.asm для просмотра. Убедимся, что файл содержит текст программы.

Левая панель				Правая панель			
Файл		Команда		Настройки		Правая панель	
< ~/work/arch-pc						< ~	
Имя		Размер		Время правки		Имя	
/..		-ВВЕРХ-		ноя 23 11:25		/..	
/lab05		2048		ноя 23 11:26		/.cache	
lab5-1.asm		2435		ноя 23 11:43		/.config	
lab5-1.asm.save		2436		ноя 23 11:32		/.emacs.d	
lab5-1.o		768		ноя 23 11:36		/.gnupg	
						/.kde4	
						/.local	
						/mozilla	

```

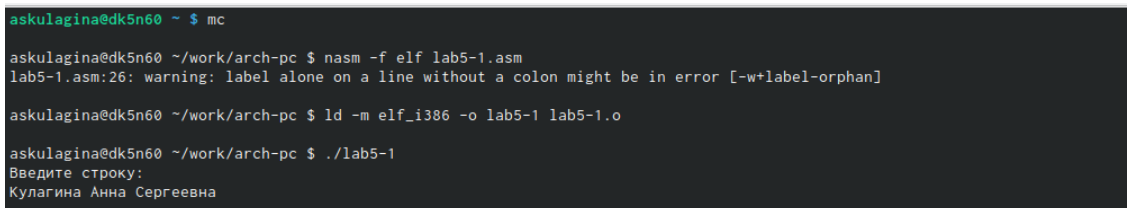
lab5-1.asm      [----] 20 L: [ 1+35 36/ 36] *(2434/2434b) <EOF>
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: 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 ; Дескриптор 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

```

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. ??)

- 3) Оттранслируем текст программы lab5-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. На запрос введем наше ФИО.

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.2)



```
askulagina@dk5n60 ~ $ mc
askulagina@dk5n60 ~/work/arch-pc $ nasm -f elf lab5-1.asm
lab5-1.asm:26: warning: label alone on a line without a colon might be in error [-w+label-orphan]
askulagina@dk5n60 ~/work/arch-pc $ ld -m elf_i386 -o lab5-1 lab5-1.o
askulagina@dk5n60 ~/work/arch-pc $ ./lab5-1
Введите строку:
Кулагина Анна Сергеевна
```

Рис. 3.2: 4.png

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.3)

4) Для вызова подпрограммы из внешнего файла используем `call`. Скачаем файл `in_out.asm` со страницы курса в ТУИС. В одной из панелей `mc` откроем каталог с файлом `lab5-1.asm`. В другой панели каталог со скаченным файлом `in_out.asm` (для перемещения между панелями используем `Tab`). Скопируем файл `in_out.asm` в каталог с файлом `lab5-1.asm` с помощью функциональной клавиши `F5`. С помощью функциональной клавиши `F6` создадим копию файла `lab5-1.asm` с именем `lab5-2.asm`. Выделим файл `lab5-1.asm`, нажмем клавишу `F6`, введем имя файла `lab5-2.asm` и нажмем клавишу `Enter`.

```
/afs/.dk.sci.pfu.edu.ru/home/a/s/askulagina/work/study/2022-2023/Архитектура
;-----
; 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 sprintLF ; 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. 3.3: 5.png

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. ??)

5)В файле lab5-2.asm заменим подпрограмму sprintLF на sprint. Создадим исполняемый файл и проверим его работу. Разница в том, что текст выводится на той же строке, где расположена фраза “Введите строку”, а не на отдельной.

```
mc [askulagina@dk2n25]:~/work/study/2022-2023/Архитектура компьюте...
../study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05/lab5-2.asm
-----
; 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 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
call quit ; вызов подпрограммы завершения
```

Ход выполнения самостоятельной работы: 1) Создадим копию файла lab5-1.asm. Внесем изменения в программу (без использования внешнего файла in_out.asm). 2) Получим исполняемый файл и проверим его работу. На приглашение введем строку введите свою фамилию. 3) Создадим копию файла lab6-2.asm. Исправим текст программы с использованием подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму: • выведем приглашение типа “Введите строку:”; • введем строку с клавиатуры; • выведем введенную строку на экран. 4) Создадим исполняемый файл и проверим его работу.

4 Выводы

Я приобрела практические навыки работы с редакторами nano и gedit. А также познакомился с Midnight Commander.

Список литературы