Отчет по лабораторной работе №5

Архитектура компьютера

Дмитрий Константинович Кобзев

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

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

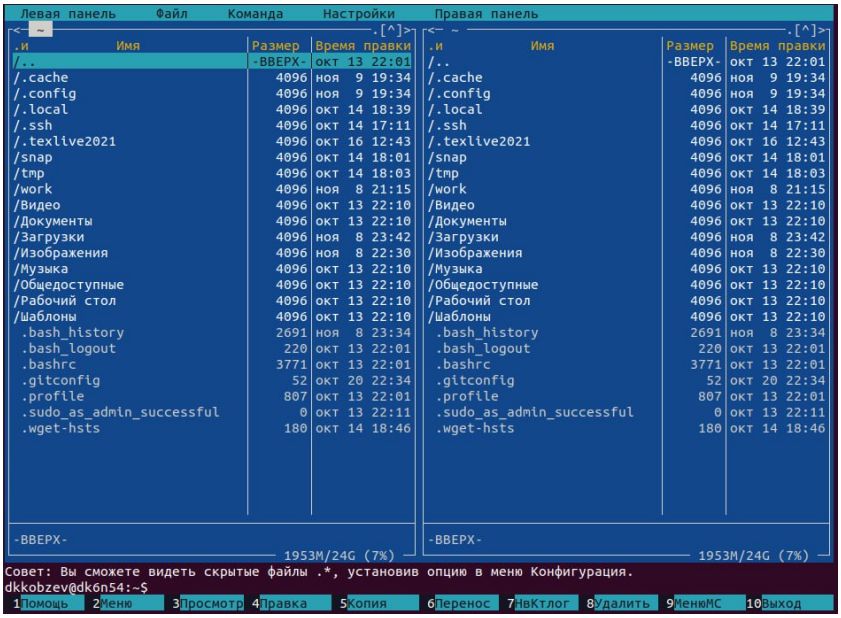
2 Задание

1. Создайте копию файла lab5-1.asm. Внесите изменения в программу (без использова- ния 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 под- программ 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.

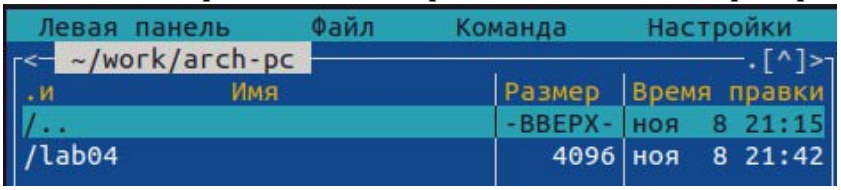
3 Выполнение лабораторной работы |

[1–6]

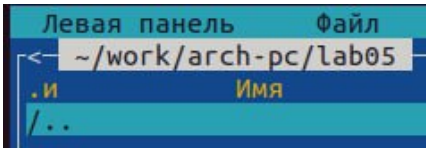
Открываем Midnight Commander.



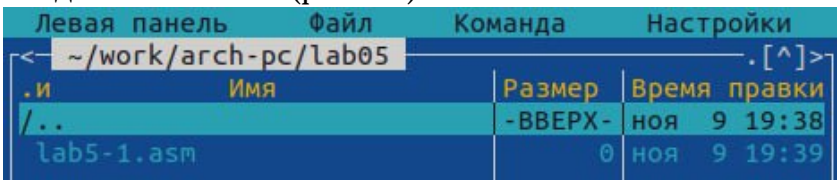
Переходим в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4 (рис. 1.2).



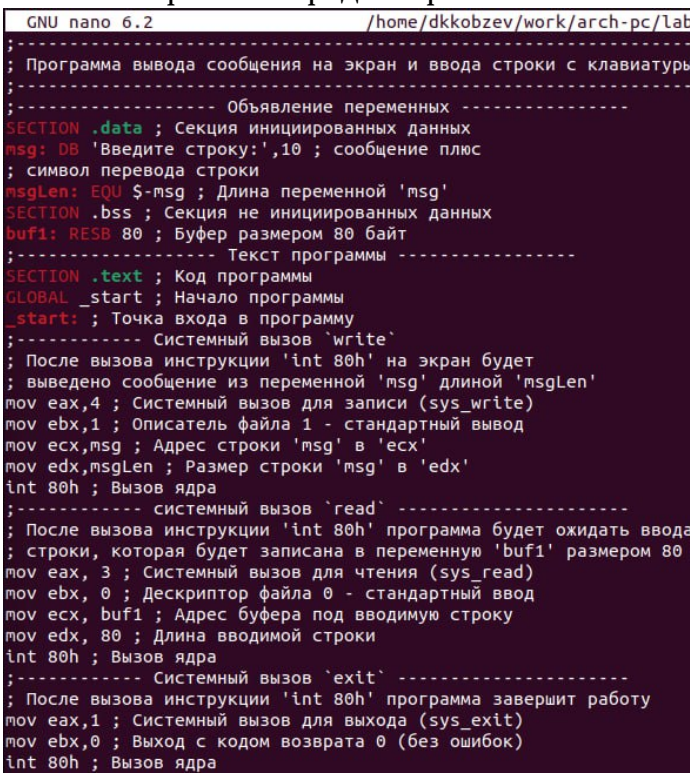
Создаем папку lab05 и переходим в созданный каталог (рис. 1.3).



Создаем файл lab5-1.asm (рис. 1.4).



Открываем файл lab5-1.asm для редактирования во встроенном редактори и вводим текст программы листинга 5.1. (рис. 1.5).



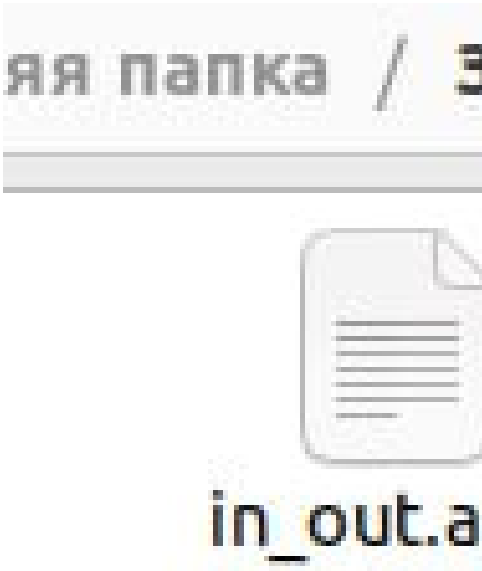
Открываем файл lab5-1.asm для просмотра и убеждаемся, что файл содержит

текст программы (рис. 1.6).

```
/home/dkkobzev/work/arch-pc/lab05/lab5-1.asm
;-------------------------------------------------------------
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-------------------------------------------------------------
;------------------ Объявление переменных ----------------
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициированных 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
```

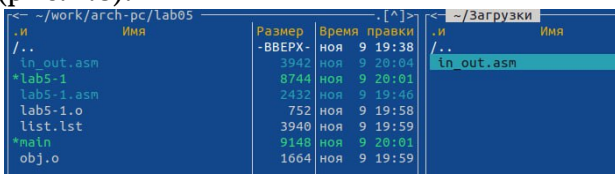
Транслируем текст программы lab5-1.asm в объектный файл. Выполняем компоновку объектного файла и запускаем получившийся исполняемый файл. Программа выводит строку ‘Введите строку:’ и ожидает ввода с клавиатуры. На запрос вводим ФИО(рис. 1.7).

```
dkkobzev@dk6n54:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-1.o -o lab5-1
dkkobzev@dk6n54:~/work/arch-pc/lab05$ ld -m elf_i386 obj.o -o main
dkkobzev@dk6n54:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Кобзев Дмитрий Константинович
```

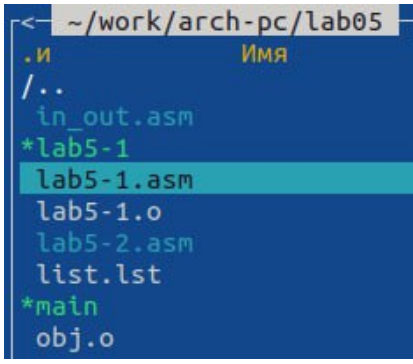


Качаем файл in_out.asm со страницы курса в ТУИС. (рис. 1.8).

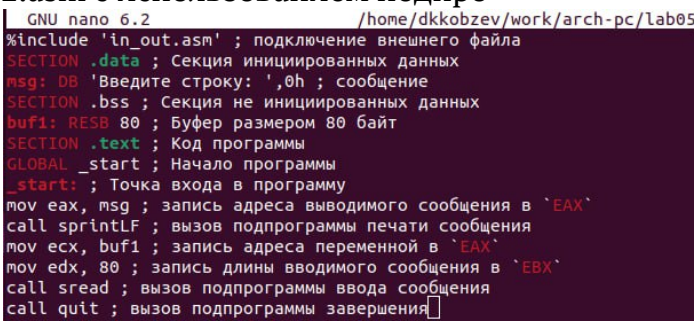
Копируем файл in_out.asm в каталог lab05 (рис. 1.9).



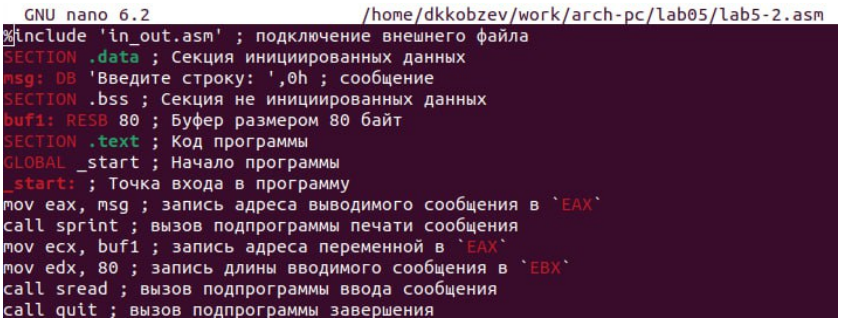
Создаем копию файла lab5-1.asm с именем lab5-2.asm. (рис. 1.10).



Исправляем текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm.(рис. 1.11).



В файле lab5-2.asm замените подпрограмму sprintLF на sprint. (рис. 1.11).


GNU nano 6.2 /home/dkkobzev/work/arch-pc/lab05/lab5-2.asm
%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
call quit ; вызов подпрограммы завершения

4 Самостоятельная работа

Задание 1. Создаем копию файла lab5-1.asm. Вносим изменения в программу (без использова- ния 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. (рис. 2.1).

```
 GNU nano 6.2                  /home/dkkobzev/work/arch-pc/lab05/lab5-1copy.asm
;-----------------------------------------------------------------------
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----------------------------------------------------------------------
;------------------- Объявление переменных ----------------
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициированных 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
mov eax,4 ; Системный вызов для выхода (sys_exit)
mov ebx,1 ; Выход с кодом возврата 0 (без ошибок)
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, buf1 ; 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
```

Задание 2. Получаем исполняемый файл и проверьте его работу. На приглашение ввести строку вводим фамилию. (рис. 2.2).

```
dkkobzev@dk6n54:~/work/arch-pc/lab05$ ./lab5
Введите строку:
Кобзев
Кобзев
```

Задание 3. Создаем копию файла lab5-2.asm. Вносим изменения в программу с использованием 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.

```
 GNU nano 6.2     /home/dkkobzev/work/arch-pc/lab05/lab5-2copy.asm
%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,4
mov ebx,1
mov ecx, buf1
int 80h
call quit ; вызов подпрограммы завершения
```

Задание 4. Получаем исполняемый файл и проверьте его работу. На приглашение ввести строку вводим фамилию.

```
dkkobzev@dk6n54:~/work/arch-pc/lab05$ ./lab5-2copy
Введите строку: Кобзев
Кобзев
```

5 Выводы

В ходе выполнения лабораторной работы мною были приобретены практические навыки работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: https://www.gnu.org/software/bash/manual/.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.