

Лабораторная работа номер 4

Палымбетов Амирхан

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7

Список иллюстраций

Список таблиц

1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера `mov` и `int`.

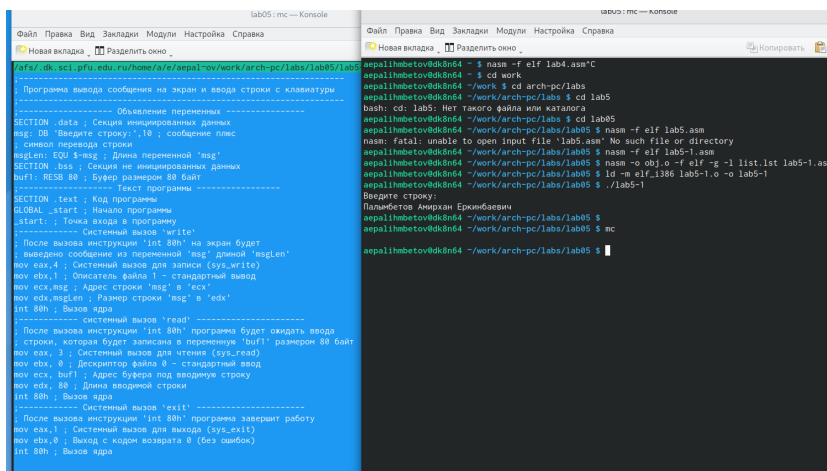
2 Задание

1. Основы работы с тс
2. Подключение внешнего файла
3. Выполнение заданий для самостоятельной работы

3 Выполнение лабораторной работы

1. Основы работы с mc:

в этом пункте мы научились использовать функциональные клавиши, а также написали



The image shows two terminal windows side-by-side. The left window displays assembly code for a program named 'lab05.asm'. The code includes sections for data (initializing a string 'msg' and a buffer 'buf'), text (program logic), and global symbols. It uses system calls like 'write' and 'read' to interact with the user. The right window shows a series of shell commands being executed in a terminal. The user attempts to assemble 'lab4.asm' and 'lab5.asm', but encounters errors because the files do not exist in the current directory. The user then navigates to the correct directory and successfully assembles 'lab5.asm' into 'lab5.o' and 'lab5-1'.

2. Подключение

внешнего файла: научились подключать внешний файл и использовать подпрограммы для более удобной работы кода `sprint` – вывод сообщения на экран, перед вызовом `sprint` в регистр `eax` необходимо записать выводимое сообщение `sprintLF` – работает аналогично `sprint`, но при выводе на экран добавляет к сообщению символ перевода строки

```
lab5-1.asm [----] 11:4:0 1+12 13/171 x(447/12220) 0032 0x020
; Проверяем наличие сообщения на запись в файл. (строка с комментарием)
%include "in_out.asm" ; Содержимое 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 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
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 записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
mov edx,buf1 ; Размер строки buf1
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
call quit ; вызов подпрограммы завершения
```

3.Выполнение самосто-

ятельной работы Изменяем код, играемся с командам, что бы в итоге получился

```
аералиhmbetov@dk8n64 ~/work/arch-pc/labs/lab05 $ nasm -f elf lab5-11.asm
аералиhmbetov@dk8n64 ~/work/arch-pc/labs/lab05 $ nasm -o obj.o -f elf -g -l list.lst lab5-11.asm
аералиhmbetov@dk8n64 ~/work/arch-pc/labs/lab05 $ ld -m elf_i386 lab5-11.o -o lab5-11
аералиhmbetov@dk8n64 ~/work/arch-pc/labs/lab05 $ ./lab5-2
Введите строку: gfa
аералиhmbetov@dk8n64 ~/work/arch-pc/labs/lab05 $ ./lab5-11
Введите строку:
Пальмбетов Амирхан
Пальмбетов Амирхан
аералиhmbetov@dk8n64 ~/work/arch-pc/labs/lab05 $
```

нужный нам результат

```
А /afs/.dk.sci.pfu.edu.ru/home/a/e/aepal~v/work/arch-pc/labs/lab05/
SECTION .data ; Секция инициализированных данных
msg: DB "Введите строку: ",10
msgLen: EQU $-msg ; Длина переменной 'msg'
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,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
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 записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
mov edx,buf1 ; Размер строки buf1
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
```

```
А lab5-22.asm [----] 41 L: 1+17 18
%include "in_out.asm"
SECTION .data ; Секция инициализированных данн
msg: DB "Введите строку: ",0h ; сообщение
SECTION .bss ; Секция не инициализированных да
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 со
call sprint ; вызов подпрограммы печати со
mov ecx,buf1 ; запись адреса переменной в
mov edx,80 ; запись длины вводимого сообщ
call sread ; вызов подпрограммы ввода сооб
mov eax,4 ; Системный вызов для записи (sy
mov ebx,1 ; Описатель файла '1' - стандарт
mov ecx,buf1 ; Адрес строки buf1 в ecx
int 80h ; Вызов ядра
call quit ; вызов подпрограммы завершения
```



```

aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab05 $ ld -m elf_i386 lab5-22.o -o lab5-22
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab05 $ nasm -o obj.o -f elf -g -l list.lst lab5-22.asm
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab05 $ ld -m elf_i386 lab5-22.o -o lab5-22
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab05 $ ./lab5-22
Введите строку: Палымбетов Амирхан
Палымбетов Амирхан
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab05 $

```

Рис7 Рис8 # Выводы

При выполнении данной лабораторной работы я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера mov и int.