Лабораторная работа номер 4

Палымбетов Амирхан

Содержание

| 1 | Цель работы | 5 |
|---|--------------------------------|----|
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 4 | Выводы | 11 |

Список иллюстраций

| 3.1 | Рис1 | | | | | | | | | | | | | | | | | 7 |
|-----|------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|----|
| 3.2 | Рис2 | | | | | | | | | | | | | | | | | 8 |
| 3.3 | Рис3 | | | | | | | | | | | | | | | | | 8 |
| 3.4 | Рис4 | | | | | | | | | | | | | | | | | ç |
| 3.5 | Рис5 | | | | | | | | | | | | | | | | | 10 |
| 3.6 | Рис6 | | | | | | | | | | | | | | | | | 10 |

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 программу ввода строки с клавиатуры, использовав при этом знания обращения с файлом из предыдущей лабораторной работы

Рис. 3.1: Рис1

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 еах необходимо записать выводимое сообщение sprintLF – работает аналогично sprint, но при выводе на экран добавляет к сообще- нию символ перевода строки

```
| Comparison to the property of the property
```

Рис. 3.2: Рис2

3.Выполнение самостоятельной работы Изменяем код, играемся с командам, что бы в итоге получился нужный нам результат

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

Рис. 3.3: Рис3

```
/afs/.dk.sci.pfu.edu.ru/home/a/e/aepal~v/work/arch-pc/labs/lab05/SECTION .data ; Секция инициированных данных msg: DB 'Bведите строку:',10 msgLen: EQU $-msg ; Длина переменной 'msg' 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,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' 5 г int 80h ; Вызов ядра Вызов для чтения (sys_read) рай mov eax, 3 ; Системный вызов для чтения (sys_read) рай mov eax, 0 ; Дескриптор файла 0 - стандартный ввод 0 c mov ecx, buf1 ; Адрес буфера под вводимую строку mov edx, 80 ; Длина вводимой строки 6 int 80h ; Вызов ядра 9 с mov eax, 4 ; Системный вызов для записи (sys_write) буф mov eax, 1 ; Описатель файла 'l' - стандартный вывод моv ecx, buf1 ; Адрес строки buf1 в ecx вызов вдра про mov edx, 1 ; Описатель файла 'l' - стандартный вывод про ex, buf1 ; Размер строки buf1 в ecx вызов ядра про mov eax, 1 ; Системный вызов для выхода (sys_exit) cmp mov ebx, 0 ; Выход с кодом возврата 0 (без ошибок) int 80h ; Вызов ядра дол
```

Рис. 3.4: Рис4

Рис. 3.5: Рис5

```
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 Beeдите строку: Палымбетов Амирхан Палымбетов Амирхан aepalihmbetov@dk8n64 -/work/arch-pc/labs/lab05 $
```

Рис. 3.6: Рис6

4 Выводы

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