

# **Шаблон отчёта по лабораторной работ №5**

**Дисциплина: архитектура компьютера**

Хоюгбан Ганчыыр Анатольевич

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работа	6

## Список иллюстраций

2.1	Команда “mc” . . . . .	6
2.2	Открытие каталога “lab05” . . . . .	6
2.3	Текст программы . . . . .	7
2.4	Наличие файла “lab5-1.asm” . . . . .	7
2.5	Исполнение файла lab5-1.asm . . . . .	8
2.6	Исполнение файла lab5-2.asm . . . . .	8
2.7	Текст программы “lab5-2.asm” . . . . .	8
2.8	Текст программы для 1 пункта . . . . .	9
2.9	Исполнение файла “lab5-1-1.asm” . . . . .	9
2.10	Текст программы для пункта 3(файл “lab-2.asm”) . . . . .	10
2.11	Исполнение программы для 3 пункта(файла “lab-2.asm”) . . . . .	10

## Список таблиц

# 1 Цель работы

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



Текст программы для вывода сообщения на экран и ввода строки с клавиатуры(рис. [fig:003])

```
;-----  
; 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
```

Рис. 2.3: Текст программы

Проверяю появился ли файл “lab5-1.asm”(рис. [fig:004])

```
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ ls  
in_out.asm lab5-1 lab5-1.asm lab5-1.o
```

Рис. 2.4: Наличие файла “lab5-1.asm”

Оттранслировал текст программы lab5-1.asm в объектный файл, выполнил компоновку и запустил получившийся файл на исполнение. Программа вывела мне “введите строку”, куда я ввел свое ФИО(рис. [fig:005]).

```

gakhoyugban@dk2n22 ~ $ cd work
gakhoyugban@dk2n22 ~/work $ cd arch-pc
gakhoyugban@dk2n22 ~/work/arch-pc $ cd lab05
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Хоюгбан Ганчыыр Анатольевич

```

Рис. 2.5: Исполнение файла lab5-1.asm

Создаю файл lab5-2.asm и изменяю в нем текст, который я приложу в следующем рисунке. Оттранслировал текст программы lab5-2.asm в объектный файл, его компоновка и отправление на исполнение файла(рис. [fig:006]).

```

gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ touch lab5-2.asm
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ mc

gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Хоюгбан Ганчыыр Анатольевич
Хоюгбан Ганчыыр Анатольевич

```

Рис. 2.6: Исполнение файла lab5-2.asm

Текст программы для вывода сообщения на экран и ввода строки с клавиатуры(файл lab5-2.asm)(рис. [fig:007]).

```

%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 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. 2.7: Текст программы “lab5-2.asm

#Задание для самостоятельной работы



Для пункта 1 я создал файл lab5-1-1.asm, где ввел текст на рисунке(рис. [fig:008]).

```
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не иницированных 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
```

Рис. 2.8: Текст программы для 1 пункта

Затем я оттранслировал текст программы lab5-1-1.asm, сделал его компоновку и отправил на исполнение. После выполнения, получив “введите строку”, по условию самостоятельной я ввел свою фамилию(пункт 2)(рис. [fig:009]).

```
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1-1.asm
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ ./lab5-1-1
Введите строку:
Хоюгбан
Хоюгбан
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $
```

Рис. 2.9: Исполнение файла “lab5-1-1.asm”

Для 3 пункта я создал файл “lab-2.asm”, для которого я ввел текст на рисунке(рис. [fig:010]).

```
%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
```

Рис. 2.10: Текст программы для пункта 3(файл “lab-2.asm”)

Оттранслировал текст программы “lab-2.asm”, сделал компоновку и отправил на исполнение. На выходе мне надо было ввести строку, куда я ввел свое ФИО. Затем я получил такую же строчку, то есть свое ФИО, а значит по условию у меня все получилось(рис. [fig:011]).

```
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
nasm: fatal: unable to open input file 'lab5-2.asm' No such file or directory
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ nasm -f elf lab-2.asm
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ ls
in_out.asm lab-2.asm lab-2.o lab5-1 lab5-1.o
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab-2 lab-2.o
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $ ./lab-2
Введите строку: Хюгбан Ганчыр Анатоьевич
gakhoyugban@dk2n22 ~/work/arch-pc/lab05 $
```

Рис. 2.11: Исполнение программы для 3 пункта(файла “lab-2.asm”)

### #Вывод

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