

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ДОКЛАД

на тему «**НАЗВАНИЕ ТЕМЫ**»

дисциплина: *Архитектура компьютера*

Студент: *Мартемьянов М.С.*

Группа: НКАбд-04-25

№ ст. билета: 1032250312

МОСКВА

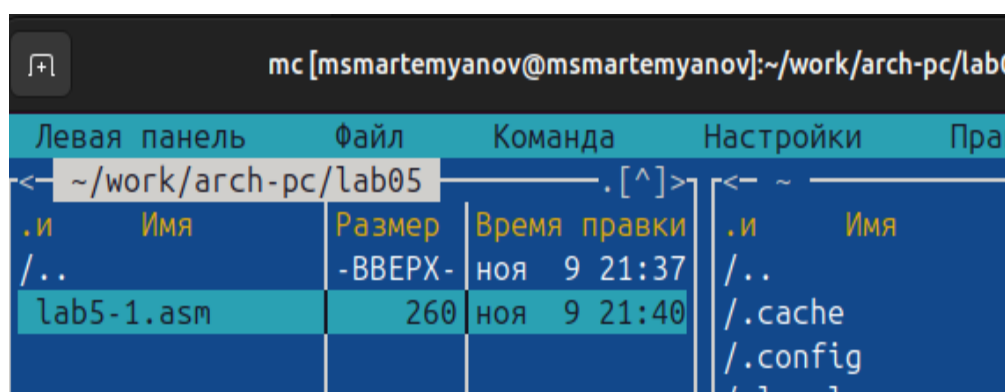
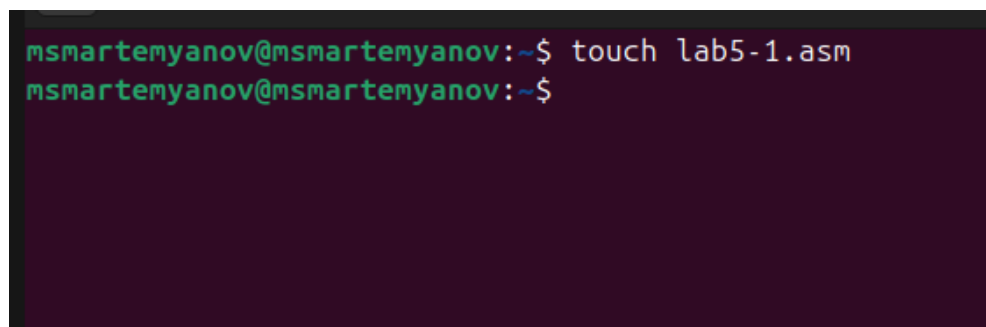
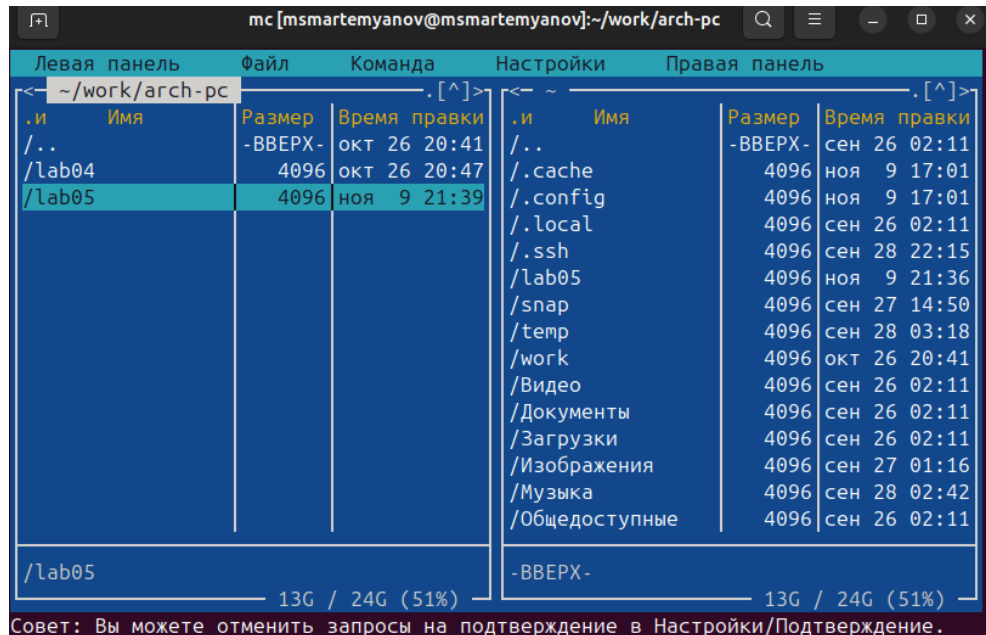
2025г.

1.Цель работы:

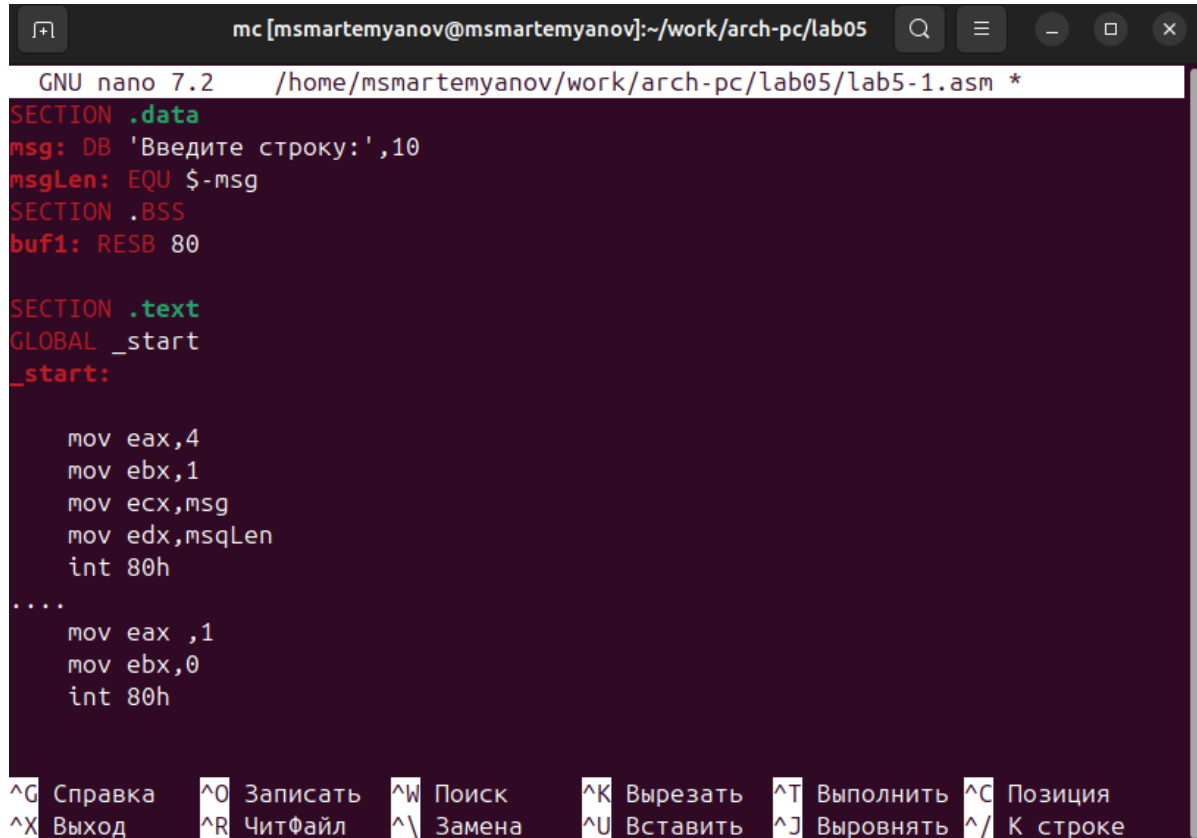
Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int.

2. Выполнение лабораторной работы

Откроем Midnight Commander, перейдём в каталог `~/work/arch-pc` созданный при выполнении лабораторной работы №4, далее создадим папку `lab05`, перейдём в него и с помощью команды `touch` создадим файл `lab5-1.asm`



С помощью функциональной клавиши F6 создадим копию файла lab5-1.asm с именем lab5-2.asm. Исправим текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm в соответствии с листингом 5.2. Создадим исполняемый файл и проверим его работу.



```
mc[msmartemyanov@msmartemyanov]:~/work/arch-pc/lab05
GNU nano 7.2 /home/msmartemyanov/work/arch-pc/lab05/lab5-1.asm *
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .BSS
buf1: RESB 80

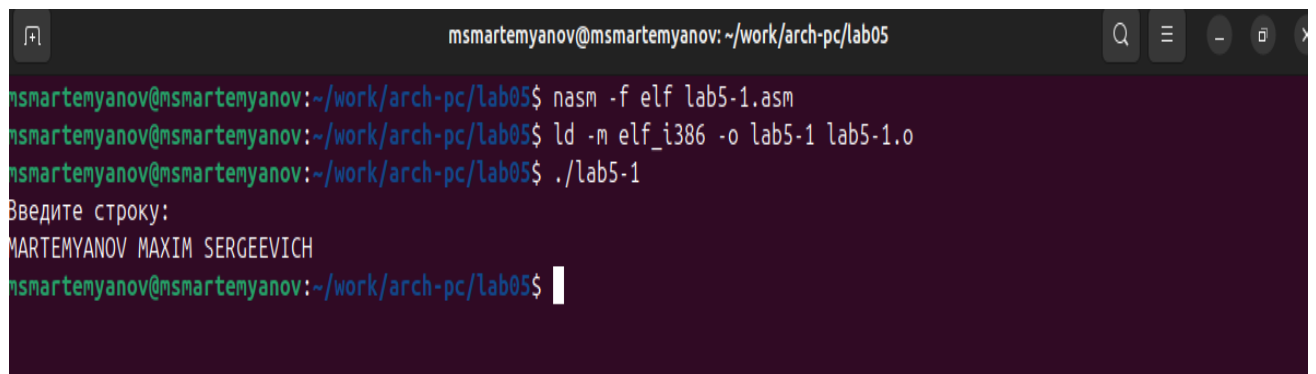
SECTION .text
GLOBAL _start
_start:

    mov eax,4
    mov ebx,1
    mov ecx,msg
    mov edx,msgLen
    int 80h

    ....
    mov eax,1
    mov ebx,0
    int 80h

^G Справка  ^O Записать  ^W Поиск    ^K Вырезать  ^T Выполнить ^C Позиция
^X Выход    ^R ЧитФайл  ^\ Замена   ^U Вставить  ^J Выводить  ^/ К строке
```

Оттранслируем текст программы lab5-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа должна вывести строку 'Введите строку:', она будет ожидать ввода с клавиатуры. На запрос введём своё ФИО.



```
msmartemyanov@msmartemyanov:~/work/arch-pc/lab05
msmartemyanov@msmartemyanov:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
msmartemyanov@msmartemyanov:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
MARTEMYANOV MAXIM SERGEEVICH
msmartemyanov@msmartemyanov:~/work/arch-pc/lab05$
```

С помощью функциональной клавиши F6 создадим копию файла lab5-1.asm с именем lab5-2.asm. Исправим текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm в соответствии с листингом 5.2. Создадим исполняемый файл и проверим его работу.

```
GNU nano 7.2 /home/msmartemyanov/work/arch-pc/lab05/lab5-2.asm *
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите строку: ',0h

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:
    mov eax, msg
    call sprintLF

    mov ecx, buf1
    mov edx, 80
    call sread

    call quit
```

```
msmartemyanov@msmartemyanov:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
msmartemyanov@msmartemyanov:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
MARTEMYANOV MAXIM SERGEEVICH
msmartemyanov@msmartemyanov:~/work/arch-pc/lab05$
```

Заметим, что при использовании в lab5-2.asm подпрограмм из внешнего файла in_out.asm запрашиваемая строка выводится сразу же, а не на следующей строке, как это было в файле lab5-1.asm

3. Задания для самостоятельной работы

1) Создадим копию файла lab5-1.asm и назовём его lab5-3.asm . Внесём 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 введенную строку на экран

```

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

SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start:

; Вывод приглашения для ввода
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

; Ввод строки с клавиатуры
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

; Сохраняем длину введенной строки для последующего вывода
mov esi, eax ; Сохраняем количество прочитанных байтов

; Вывод введенной строки на экран
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,buf1 ; Адрес буфера с введенной строкой
mov edx,esi ; Количество байтов для вывода (длина введенной строки)
int 80h ; Вызов ядра

; Завершение программы
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h

```

2)Создали файл lab5-2, выполнили компоновку объектного файла и запустили получившийся исполняемый файл. Выполнение компоновки объектного файла и его запуск текст программы файла lab5-3.asm 2) Создадим копию файла lab5-2.asm

с названием lab5-4.asm . Исправим текст программы 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 введенную строку на экран. Создадим исполняемый файл и проверим его работу. Выполнение компоновки объектного файла и его запуск текст программы файла lab5-4.asm

```
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',10; 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
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Вывод Приобрел практические навыки работы в Midnight Commander. Освоил инструкции языка ассемблера mov и int.

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).