

# Отчёт по лабораторной работе 4

## Архитектура компьютера

Тарасова Анна Викторовна

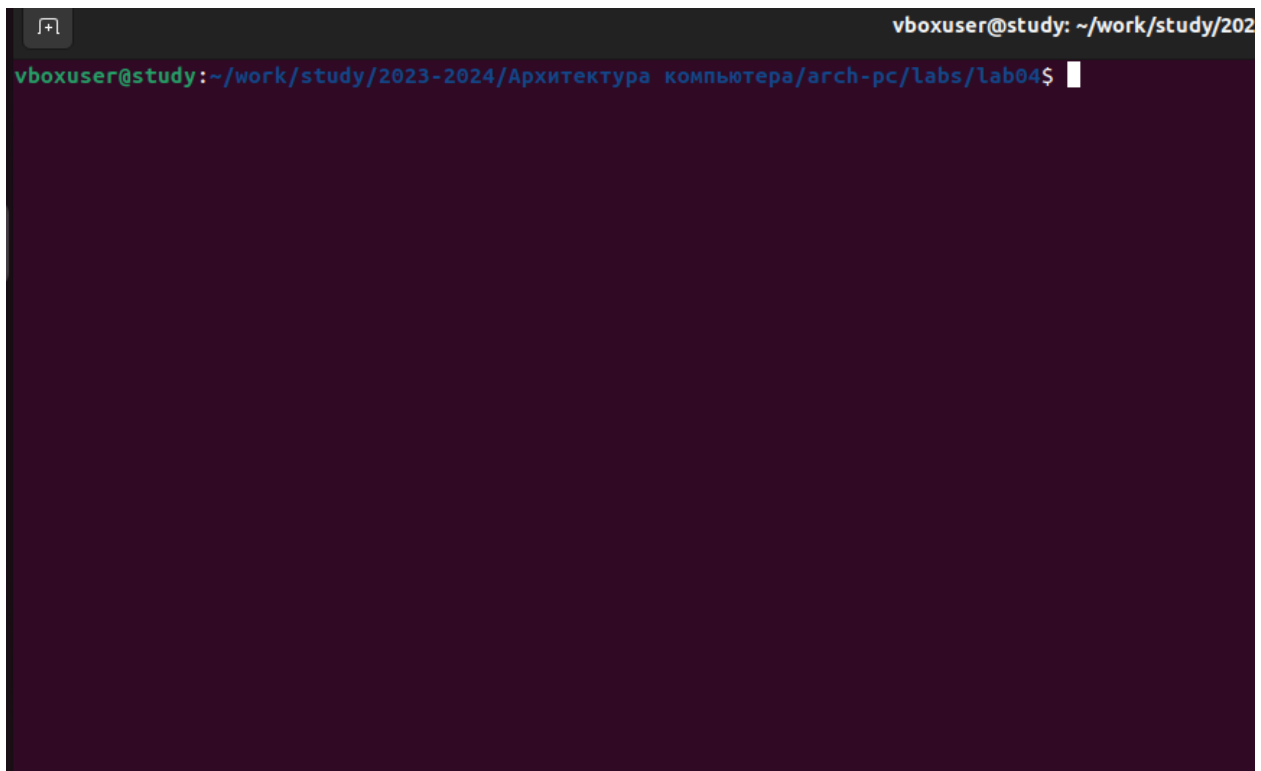
### Содержание

### Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

### Выполнение лабораторной работы

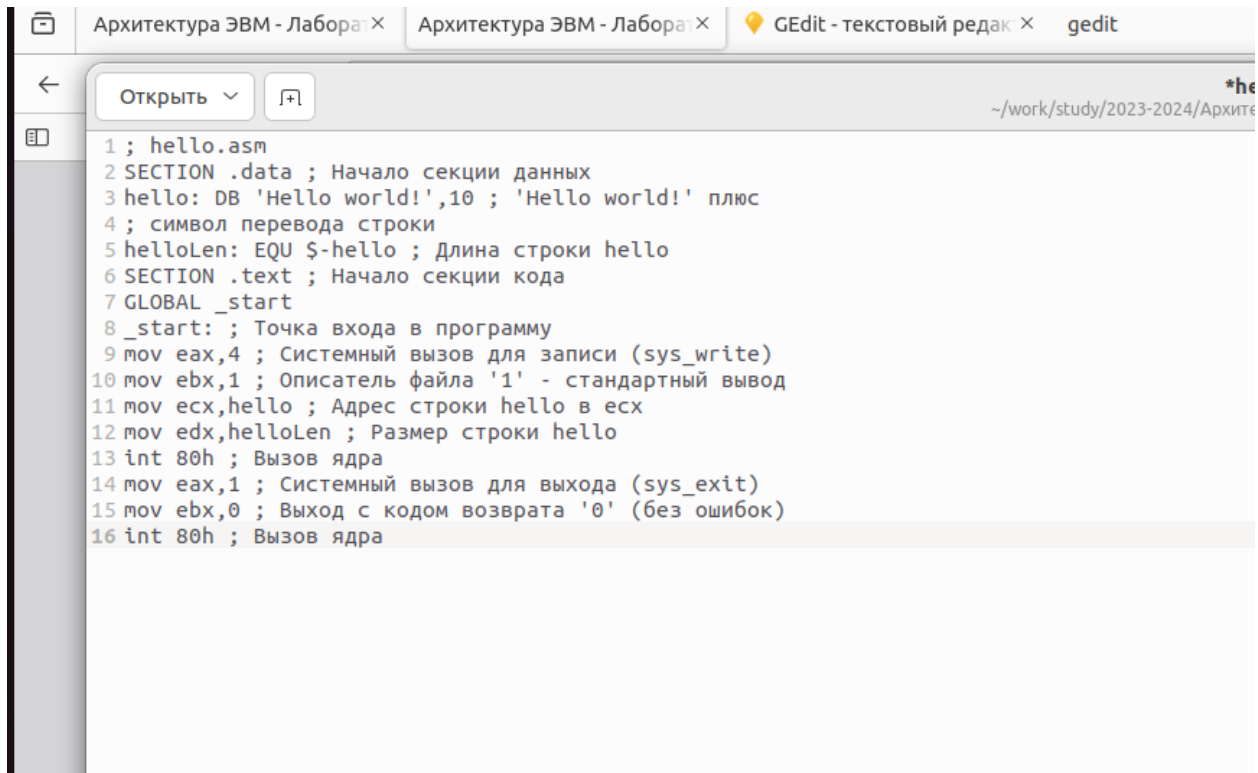
1. С помощью утилиты `cd` перемещаюсь в каталог в котором буду работать.

A screenshot of a terminal window with a dark background. The title bar at the top shows a window icon and the text 'vboxuser@study: ~/work/study/202'. The terminal prompt is 'vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04\$' followed by a white cursor. The rest of the terminal area is empty.

```
vboxuser@study: ~/work/study/202
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
```

Затем с помощью утилиты touch создаю пустой текстовый файл hello.asm,открываю его и заполняю(вставляю программу для вывода «Hello word!»)

```
vboxuser@study: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ touch hello.asm
```



```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для записи (sys_write)
10 mov ebx,1 ; Описатель файла '1' - стандартный вывод
11 mov ecx,hello ; Адрес строки hello в ecx
12 mov edx,helloLen ; Размер строки hello
13 int 80h ; Вызов ядра
14 mov eax,1 ; Системный вызов для выхода (sys_exit)
15 mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
16 int 80h ; Вызов ядра
```

Потом я превращаю текст программы в код с помощью транслятора NASM и проверяю правильность выполнения команды с помощью ls.

```
vboxuser@study: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ touch hello.asm
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ gedit hello.asm
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ nasm -f elf hello.asm
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm  hello.o
```

Следующим шагом было скомпилировать файл hello.asm в файл obj.o и проверка с помощью ls.

```
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
```

Затем передаю объектный файл hello.o на обработку компоновщику LD , чтобы получить исполняемый файл hello и выполняю команду для получения файла main. Запускаю на выполнение созданный файл исполняемый файл hello.

```
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ld -m elf_i386 hello.o -o hello
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ld -m elf_i386 obj.o -o main
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello hello.asm hello.o list.lst main obj.o
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ./hello
Hello world!
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
```

## Выполнение заданий для самостоятельной работы.

С помощью утилиты cp создаю копию файла hello.asm с именем lab4.asm .

```
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
cp hello.asm lab4.asm
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
ls
hello      hello.o  list.lst  obj.o      report
hello.asm  lab4.asm main      presentation
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
```

С помощью текстового редактора открываю файл и вношу изменения в программу так , чтоб она выводила мои имя и фамилию.

```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Tarasova Anna!',10 ; 'мой текст!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для записи (sys_write)
10 mov ebx,1 ; Описатель файла '1' - стандартный вывод
11 mov ecx,hello ; Адрес строки hello в ecx
12 mov edx,helloLen ; Размер строки hello
13 int 80h ; Вызов ядра
14 mov eax,1 ; Системный вызов для выхода (sys_exit)
15 mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
16 int 80h ; Вызов ядра
```

Компилирую текст программы в объектный файл и проверяю ,что файл создан. Затем запускаю исполняемый файл lab4 .

```

bash: ./lab4: Нет такого файла или каталога
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ld -m elf_i386 lab4.o -o lab4
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o presentation report
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ./lab4
Tarasova Anna!
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$

```

Удаляю лишние файлы в текущем каталоге.

```

vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ rm hello hello.o lab4 lab4.o list.lst main obj.o
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm lab4.asm presentation report
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ █

```

Добавляю файлы на Github.

```

vboxuser@study:~$ cd work/study/2023-2024/"Архитектура компьютера"/arch-pc
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -m "Add files for lab04"
[master 8e3af9f] Add files for lab04
15 files changed, 197 insertions(+), 158 deletions(-)
create mode 100755 lab04/hello
create mode 100644 lab04/hello.asm
create mode 100644 lab04/hello.o
create mode 100755 lab04/lab4
create mode 100644 lab04/lab4.asm
create mode 100644 lab04/lab4.o
create mode 100644 lab04/list.lst
create mode 100755 lab04/main
create mode 100644 lab04/obj.o
rewrite labs/lab02/report/Makefile (85%)
delete mode 100644 labs/lab03/report/image/placeimg_800_600_tech.jpg
rewrite labs/lab03/report/report.md (70%)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 30, готово.
Подсчет объектов: 100% (29/29), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (21/21), готово.
Запись объектов: 100% (21/21), 6.06 Киб | 3.03 Миб/с, готово.
Всего 21 (изменений 9), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (9/9), completed with 4 local objects.
To github.com:Anna142004/study_2023-2024_arch-pc.git
   b2ddb21..8e3af9f  master -> master
vboxuser@study:~/work/study/2023-2024/Архитектура компьютера/arch-pc$

```

## Вывод

При выполнении данной лабораторной работы я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.