

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ
НАРОДОВ**

**Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4**

дисциплина: Архитектура компьютера

Студент: Абронина А.К.

Группа: НКАбд-01-24

С/Б 1132246717

МОСКВА

2024г

Цель работы

Цель данной лабораторной работы - освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.

Выполнение лабораторной работы

Создание программы Hello world!

Создаю в текущем каталоге пустой текстовый файл hello.asm с помощью утилиты touch.

```
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs$ cd lab04
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$ touch hello.asm
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$ ls
hello.asm presentation report
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$
```

Рис.1 Создание пустого файла

Открываю созданный файл в текстовом редакторе mousepad.

```
hello.asm presentation report
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$ mousepad hello.asm
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$
```

Рис. 2 Открытие файла в текстовом редакторе

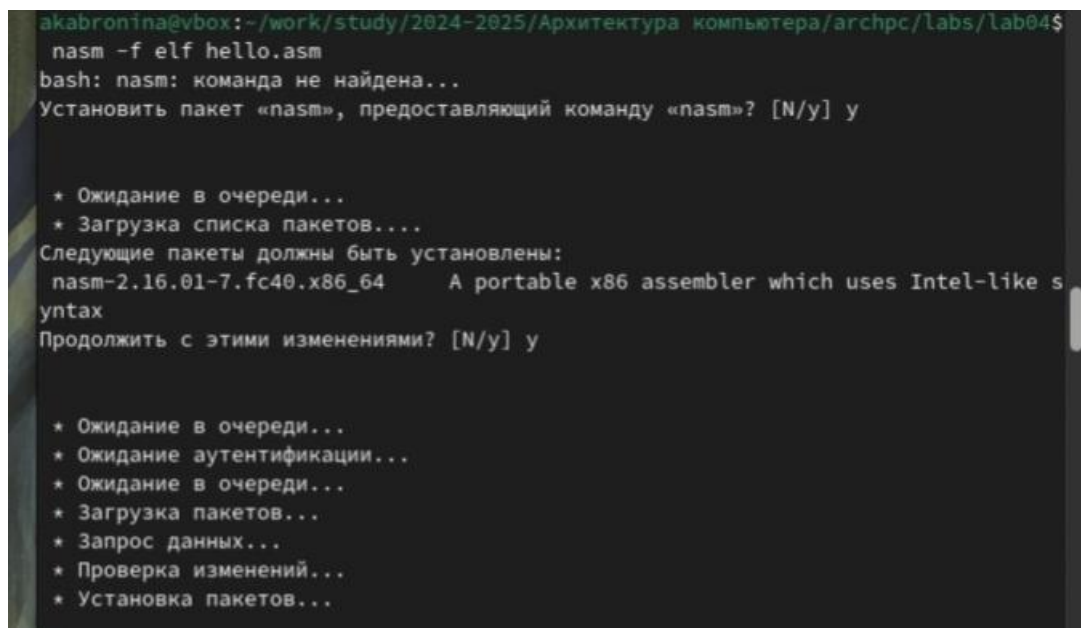
Заполняю файл, вставляя в него программу для вывода "Hello word!"

```
Файл  Правка  Поиск  Просмотр  Документ  Помощь
|; hello.asm
SECTION .data ; Начало секции данных
hello: DB 'Hello world!',10 ; 'Hello world!' плюс
; символ перевода строки
helloLen: EQU $-hello ; Длина строки hello
SECTION .text ; Начало секции кода
GLOBAL _start
_start: ; Точка входа в программу
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,hello ; Адрес строки hello в ecx
mov edx,helloLen ; Размер строки hello
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.3 Заполнение файла

Работа с транслятором NASM

Превращаю текст программы для вывода "Hello world!" в объектный код с помощью транслятора NASM, используя команду `nasm -f elf hello.asm`, ключ `-f` указывает транслятору `nasm`, что требуется создать бинарный файл в формате ELF. Далее проверяю правильность выполнения команды с помощью утилиты `ls`: действительно, создан файл "hello.o".



```
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$  
nasm -f elf hello.asm  
bash: nasm: команда не найдена...  
Установить пакет «nasm», предоставляющий команду «nasm»? [N/y] y  
  
* Ожидание в очереди...  
* Загрузка списка пакетов...  
Следующие пакеты должны быть установлены:  
nasm-2.16.01-7.fc40.x86_64    A portable x86 assembler which uses Intel-like s  
yntax  
Продолжить с этими изменениями? [N/y] y  
  
* Ожидание в очереди...  
* Ожидание аутентификации...  
* Ожидание в очереди...  
* Загрузка пакетов...  
* Запрос данных...  
* Проверка изменений...  
* Установка пакетов...
```

Рис. 4 Компиляция текста программы

Работа с расширенным синтаксисом командной строки NASM

Ввожу команду, которая скомпилирует файл `hello.asm` в файл `obj.o`, при этом в файл будут включены символы для отладки (ключ `-g`), также с помощью ключа `-l` будет создан файл листинга `list.lst`. Далее проверяю с помощью утилиты `ls` правильность выполнения команды.

```
ls
hello.asm hello.o presentation report
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$
nasm -o obj.o -f elf -g -l list.lst hello.asm
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$
ls
hello.asm hello.o list.lst obj.o presentation report
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$
```

Рис. 5 Компиляция текста программы

Работа с компоновщиком LD

Передаю объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello. Ключ -o задает имя создаваемого исполняемого файла. Далее проверяю с помощью утилиты ls правильность выполнения команды.

```
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$
ld -m elf_i386 hello.o -o hello
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$
ls
hello hello.asm hello.o list.lst obj.o presentation report
```

Рис. 6 Передача объектного файла на обработку компоновщику

Выполняю следующую команду. Исполняемый файл будет иметь имя main, т.к. после ключа -o было задано значение main. Объектный файл, из которого собран этот исполняемый файл, имеет имя obj.o

```
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$
ld -m elf_i386 obj.o -o main
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$
ls
hello hello.asm hello.o list.lst main obj.o presentation report
```

Рис. 7 Передача объектного файла на обработку компоновщику

Запуск исполняемого файла

Запускаю на выполнение созданный исполняемый файл hello.

```
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$
./hello
Hello world!
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$
```

Рис. 8 Запуск исполняемого файла

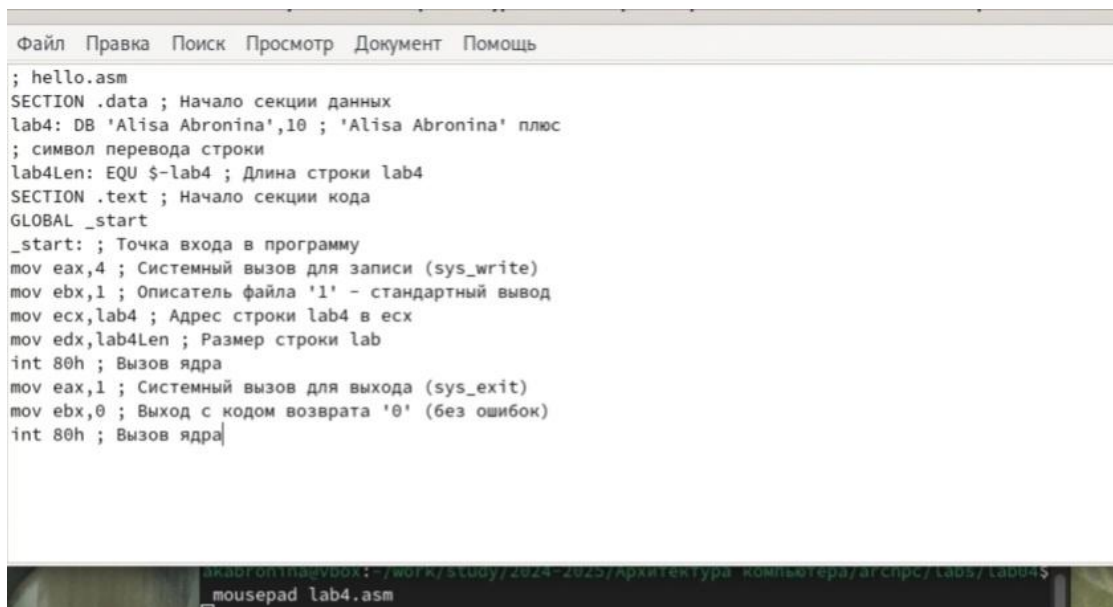
Выполнение заданий для самостоятельной работы.

С помощью утилиты `cp` создаю в текущем каталоге копию файла `hello.asm` с именем `lab4.asm`.

```
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$  
cp hello.asm lab4.asm  
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$  
ls  
hello      hello.o  list.lst  obj.o      report  
hello.asm  lab4.asm  main      presentation
```

Рис. 9 Создание копии файла

С помощью текстового редактора `mousepad` открываю файл `lab4.asm` и вношу изменения в программу так, чтобы она выводила мои имя и фамилию.



```
Файл  Правка  Поиск  Просмотр  Документ  Помощь  
; hello.asm  
SECTION .data ; Начало секции данных  
lab4: DB 'Alisa Abronina',10 ; 'Alisa Abronina' плюс  
; символ перевода строки  
lab4Len: EQU $-lab4 ; Длина строки lab4  
SECTION .text ; Начало секции кода  
GLOBAL _start  
_start: ; Точка входа в программу  
mov eax,4 ; Системный вызов для записи (sys_write)  
mov ebx,1 ; Описатель файла '1' - стандартный вывод  
mov ecx,lab4 ; Адрес строки lab4 в ecx  
mov edx,lab4Len ; Размер строки lab  
int 80h ; Вызов ядра  
mov eax,1 ; Системный вызов для выхода (sys_exit)  
mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)  
int 80h ; Вызов ядра  
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$  
mousepad lab4.asm
```

Рис. 10 Изменение программы

Компилирую текст программы в объектный файл. Проверяю с помощью утилиты `ls`, что файл `lab4.o` создан.

```
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$  
nasm -f elf lab4.asm  
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$  
ls  
hello      hello.o  lab4.o    main      presentation  
hello.asm  lab4.asm  list.lst  obj.o      report
```

Рис. 11 Компиляция текста программы

Передаю объектный файл lab4.o на обработку компоновщику LD, чтобы получить исполняемый файл lab4.

```
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$  
ld -m elf_i386 lab4.o -o lab4  
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$  
ls  
hello      hello.o  lab4.asm  list.lst  obj.o      report  
hello.asm  lab4     lab4.o    main      presentation
```

Рис. 12 Передача объектного файла на обработку компоновщику

Запускаю исполняемый файл lab4, на экран действительно выводятся мои имя и фамилия.

```
hello.asm  lab4     lab4.o    main      pres  
akabronina@vbox:~/work/study/2024-2025/Архит  
./lab4  
Alisa Abronina  
akabronina@vbox:~/work/study/2024-2025/Архит
```

Рис 13 Запуск исполняемого файла

С помощью команд git add . и git commit добавляю файлы на GitHub, комментируя действие как добавление файлов для лабораторной работы №4.

```
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$  
git add .  
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$  
git commit -m "Add files"  
[master 27aale9] Add files  
9 files changed, 49 insertions(+)  
create mode 100755 labs/lab04/hello  
create mode 100644 labs/lab04/hello.asm  
create mode 100644 labs/lab04/hello.o  
create mode 100755 labs/lab04/lab4  
create mode 100644 labs/lab04/lab4.asm  
create mode 100644 labs/lab04/lab4.o
```

Рис. 14 Добавление файлов на GitHub]

Отправляю файлы на сервер с помощью команды git push.

```
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$  
git push  
Перечисление объектов: 16, готово.  
Подсчет объектов: 100% (16/16), готово.  
При сжатии изменений используется до 2 потоков  
Сжатие объектов: 100% (13/13), готово.  
Запись объектов: 100% (13/13), 3.35 КиБ | 686.00 КиБ/с, готово.  
Total 13 (delta 7), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (7/7), completed with 2 local objects.  
To github.com:akabronina/study_2024-2025_arh-pc.git  
    ef64ff7..27aale9  master -> master  
akabronina@vbox:~/work/study/2024-2025/Архитектура компьютера/archpc/labs/lab04$
```

Рис. 15 Отправка файлов

Выводы

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