

# Отчет по лабораторной работе №4

Архитектура компьютера

Копылова Виктория Валерьевна

## Содержание

Цель работы .....	1
Задание.....	1
Теоретическое введение.....	1
Выполнение лабораторной работы.....	2
Выводы .....	5
Список литературы .....	5

## Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## Задание

1. В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создайте копию файла `hello.asm` с именем `lab4.asm`
2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы `lab4.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы `hello.asm` и `lab4.asm` в Ваш локальный репозиторий в каталог `~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/`. Загрузите файлы на Github

## Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. [-@tbl:std-dir] приведено краткое описание стандартных каталогов Unix.

## Описание некоторых каталогов файловой системы GNU Linux {#tbl:std-dir}

Имя

катало

га

Описание каталога

---

/	Корневая директория, содержащая всю файловую
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [ @tanenbaum\_book\_modern-os\_ru; @robbins\_book\_bash\_en; @zarrelli\_book\_mastering-bash\_en; @newham\_book\_learning-bash\_en ].

## Выполнение лабораторной работы

Создадим каталог для работы с программами на языке ассемблера NASM (рис. [-@fig:001]):

```
vvkopihlova@dk8n53 ~ $ mkdir -p ~/work/arch-pc/lab04
```

Перейдем в созданный каталог (рис. [-@fig:002]):

```
vvkopihlova@dk8n53 ~ $ cd ~/work/arch-pc/lab04
```

Создадим текстовый файл с именем hello.asm (рис. [-@fig:003]):

```
vvkopihlova@dk8n53 ~/work/arch-pc/lab04 $ touch hello.asm
```

Откроем этот файл с помощью любого текстового редактора, например, gedit (рис. [-@fig:004]):

```
vvkopihlova@dk8n53 ~/work/arch-pc/lab04 $ gedit hello.asm
```

И введем в него следующий текст (рис. [-@fig:005]):

```
; 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 есх
    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
```

NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать (рис. [-@fig:006]):

```
vvkopihirova@dk8n53 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
```

С помощью команды ls проверим, что объектный файл был создан (рис. [-@fig:007]):

```
vvkopihirova@dk8n53 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
```

Выполним следующую команду (рис. [-@fig:008]):

```
vvkopihirova@dk8n53 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
vvkopihirova@dk8n53 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
```

Как видно из схемы на рис. 4.3, чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику (рис. [-@fig:009]):

```
vvkopihirova@dk8n53 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
vvkopihirova@dk8n53 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Выполним следующую команду (рис. [-@fig:010]):

```
vvkopihirova@dk8n53 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
```

Формат командной строки LD можно увидеть, набрав `ld --help` (рис. [-@fig:011]):

```
vvkopihlova@dk8n53 ~/work/arch-pc/lab04 $ ld --help
Использование ld [параметры] файл...
Параметры:
  -a КЛЮЧЕВОЕ СЛОВО                Управление общей библиотекой для совместимости с HP/UX
  -A АРХИТЕКТУРА, --architecture АРХИТЕКТУРА  Задать архитектуру
  -b ЦЕЛЬ, --format ЦЕЛЬ            Задать цель для следующих входных файлов
  -c ФАЙЛ, --mri-script ФАЙЛ
```

Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге, можно, набрав в командной строке (рис. [-@fig:012]):

```
vvkopihlova@dk8n53 ~/work/arch-pc/lab04 $ ./hello
Hello world!
```

#### 4.5. Задание для самостоятельной работы

1. В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создайте копию файла `hello.asm` с именем `lab4.asm` (рис. [-@fig:013]):

```
vvkopihlova@dk8n53 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
vvkopihlova@dk8n53 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
```

2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем (рис. [-@fig:014]):

```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Копылова Виктория',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
```

3. Оттранслируйте полученный текст программы lab4.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл (рис. [-@fig:015]):

```
vvkopihlova@dk8n53 ~/work/arch-pc/lab04 $ gedit lab4.asm
vvkopihlova@dk8n53 ~/work/arch-pc/lab04 $ nasm -o Kopylova.o -f elf -g -l list2.lst lab4.asm
vvkopihlova@dk8n53 ~/work/arch-pc/lab04 $ ld -m elf_i386 Kopylova.o -o Kopylova
vvkopihlova@dk8n53 ~/work/arch-pc/lab04 $ ./Kopylova
Копылова Виктория
vvkopihlova@dk8n53 ~/work/arch-pc/lab04 $
```

4. Скопируйте файлы hello.asm и lab4.asm в Ваш локальный репозиторий в каталог ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/. Загрузите файлы на Github (рис. [-@fig:016]):

```
vvkopihlova@dk8n53 ~/work/arch-pc/lab04 $ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04
vvkopihlova@dk8n53 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello.asm lab4.asm presentation report
vvkopihlova@dk8n53 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04 $ git add .
vvkopihlova@dk8n53 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04 $ git commit -am 'feat(main): added files lab4'
[master 70c21f2] feat(main): added files lab4
 2 files changed, 32 insertions(+)
 create mode 100644 labs/lab04/hello.asm
 create mode 100644 labs/lab04/lab4.asm
vvkopihlova@dk8n53 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04 $ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 963 байта | 963.00 КиБ/с, готово.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:vviktoriaa1/study_2024-2025_arh--pc.git
 2ad4bc8..70c21f2 master -> master
```

## Выводы

Я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## Список литературы