

Отчет по лабораторной работе №4

Дисциплина: архитектура компьютера

Лопатин Павел Юрьевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение заданий для самостоятельной работы	9
4	Выводы	11

Список иллюстраций

2.1	Создание каталога	6
2.2	Файл hello.asm	6
2.3	Компиляция файла	7
2.4	Компиляция файла	7
2.5	Компоновка файла	7
2.6	Компоновка файла	7
2.7	Запуск файла	8
3.1	Копирование файла	9
3.2	Изменение программы	9
3.3	Компоновка программы	10
3.4	Исполнение программы	10
3.5	Копирование файла в репозиторий	10
3.6	Загрузка на GitHub	10

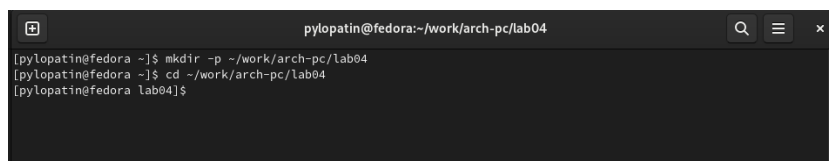
Список таблиц

1 Цель работы

Цель данной лабораторной работы - освоение процедуры компиляции и сборки программ, написанных на языке ассемблер NASM.

2 Выполнение лабораторной работы

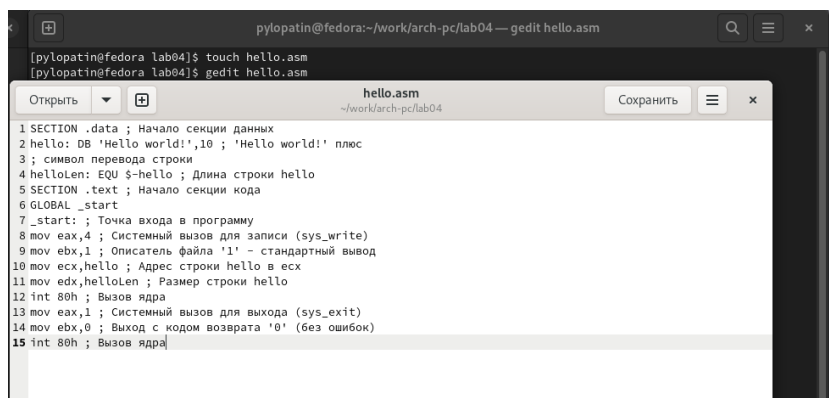
Создаю рабочий каталог arch-pc/lab04 и перехожу в неё. (рис. [2.1]).



```
pylopatin@fedora:~/work/arch-pc/lab04
[pylopatin@fedora ~]$ mkdir -p ~/work/arch-pc/lab04
[pylopatin@fedora ~]$ cd ~/work/arch-pc/lab04
[pylopatin@fedora lab04]$
```

Рис. 2.1: Создание каталога

Открываю файл hello.asm с помощью текстового редактора и набираю программу вывода hello world(рис. [2.2]).



```
pylopatin@fedora:~/work/arch-pc/lab04 — gedit hello.asm
[pylopatin@fedora lab04]$ touch hello.asm
[pylopatin@fedora lab04]$ gedit hello.asm

hello.asm
~/work/arch-pc/lab04

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

Рис. 2.2: Файл hello.asm

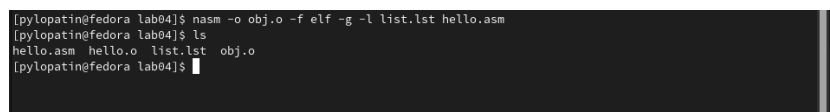
Компилирую файл программы в объектный файл и проверяю выполнение. (рис. [2.3]).



```
pylopatin@fedora:~/work/arch-pc/lab04
[pylopatin@fedora lab04]$ nasm -f elf hello.asm
[pylopatin@fedora lab04]$ ls
hello.asm  hello.o
[pylopatin@fedora lab04]$
```

Рис. 2.3: Компиляция файла

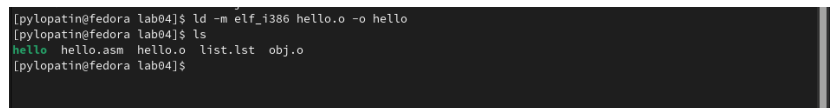
Выполняю компиляцию файла с помощью расширенного синтаксиса командной строки NASM, и проверяю выполнение, используя команду `ls`. (рис. [2.4]).



```
[pylopatin@fedora lab04]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[pylopatin@fedora lab04]$ ls
hello.asm  hello.o  list.lst  obj.o
[pylopatin@fedora lab04]$
```

Рис. 2.4: Компиляция файла

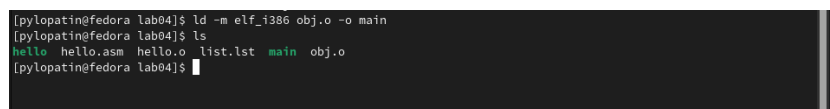
Передаю файл на обработку компоновщику LD и проверяю выполнение. (рис. [2.5]).



```
[pylopatin@fedora lab04]$ ld -m elf_i386 hello.o -o hello
[pylopatin@fedora lab04]$ ls
hello  hello.asm  hello.o  list.lst  obj.o
[pylopatin@fedora lab04]$
```

Рис. 2.5: Компоновка файла

Выполняю команду: создаю исполняемый файл `main` (рис. [2.6]).



```
[pylopatin@fedora lab04]$ ld -m elf_i386 obj.o -o main
[pylopatin@fedora lab04]$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
[pylopatin@fedora lab04]$
```

Рис. 2.6: Компоновка файла

- Исполняемый файл будет иметь имя `main`. Запуская ее мы запускаем программу. А объектный файл имеет имя `obj.o`.

Запускаю исполняемый файл (рис. [2.7]).

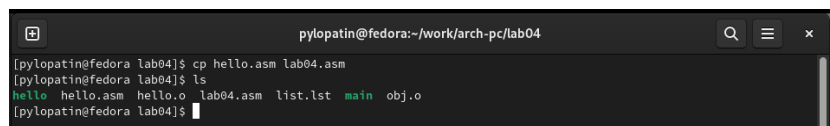
A terminal window with a dark background. The prompt is [pylopatin@fedora lab04]\$. The command ./hello is entered. The output is Hello world!. The prompt is [pylopatin@fedora lab04]\$.

```
[pylopatin@fedora lab04]$ ./hello
Hello world!
[pylopatin@fedora lab04]$
```

Рис. 2.7: Запуск файла

3 Выполнение заданий для самостоятельной работы

Копирую файл `hello.asm` и изменяю имя на `lab04.asm` и проверяю выполнение. (рис. [3.1]).



```
pylopatin@fedora:~/work/arch-pc/lab04
[pylopatin@fedora lab04]$ cp hello.asm lab04.asm
[pylopatin@fedora lab04]$ ls
hello  hello.asm  hello.o  lab04.asm  list.lst  main  obj.o
[pylopatin@fedora lab04]$
```

Рис. 3.1: Копирование файла

Открываю файл и редактирую программу так, чтобы она вывела мое имя и фамилию (рис. [3.2]).



```
[pylopatin@fedora lab04]$ gedit lab04.asm
lab04.asm
~/work/arch-pc/lab04
Сохранить

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

Рис. 3.2: Изменение программы

Транслирую текст программы в объектный файл, выполняю компоновку объектного файла. (рис. [3.3]).

```

[pylopatin@fedora lab04]$ nasm -f elf lab04.asm
[pylopatin@fedora lab04]$ ld -m elf_i386 lab04.o -o lab04
[pylopatin@fedora lab04]$ ls
hello    hello.o  lab04    lab04.asm  lab04.o  list.lst  main  obj.o
[pylopatin@fedora lab04]$

```

Рис. 3.3: Компиляция программы

Запускаю получившийся исполняемый файл (рис. [3.4]).

```

[pylopatin@fedora lab04]$ ./lab04
Лопатин Павел
[pylopatin@fedora lab04]$

```

Рис. 3.4: Исполнение программы

Копирую файлы hello.asm и lab04.asm локальный репозиторий в lab04. (рис. [3.5]).

```

pylopatin@fedora:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04
[pylopatin@fedora lab04]$ cp hello.asm lab04.asm ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04
[pylopatin@fedora lab04]$ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04
[pylopatin@fedora lab04]$ ls
hello.asm  lab04.asm  presentation  report
[pylopatin@fedora lab04]$

```

Рис. 3.5: Копирование файла в репозиторий

Загружаю файла на GitHub (рис. [3.6]).

```

[pylopatin@fedora arch-pc]$ git add .
[pylopatin@fedora arch-pc]$ git commit -am 'feat(main): add files lab04'
[master 53ce517] feat(main): add files lab04
15 files changed, 148 insertions(+), 119 deletions(-)
create mode 100644 labs/lab04/report/image/1.png
create mode 100644 labs/lab04/report/image/10.png
create mode 100644 labs/lab04/report/image/11.png
create mode 100644 labs/lab04/report/image/12.png
create mode 100644 labs/lab04/report/image/2.png
create mode 100644 labs/lab04/report/image/3.png
create mode 100644 labs/lab04/report/image/4.png
create mode 100644 labs/lab04/report/image/5.png
create mode 100644 labs/lab04/report/image/6.png
create mode 100644 labs/lab04/report/image/7.png
create mode 100644 labs/lab04/report/image/8.png
create mode 100644 labs/lab04/report/image/9.png
delete mode 100644 labs/lab04/report/image/placeimg_800_600_tech.jpg
delete mode 100644 labs/lab04/report/report.md
create mode 100644 labs/lab04/report/Л04_Лопатин_отчет.md
[pylopatin@fedora arch-pc]$ git push
Перечисление объектов: 30, готово.
Подсчет объектов: 100% (30/30), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (25/25), готово.
Запись объектов: 100% (25/25), 324.37 Киб | 1.89 Мб/с, готово.
Всего 25 изменений (6), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
To github.com:DubinaDD/study_2023-2024_arh-pc.git
  5932c9e..53ce517  master -> master
[pylopatin@fedora arch-pc]$

```

Рис. 3.6: Загрузка на GitHub

4 Выводы

Я освоил процедуры компиляции и сборки программ, написанных на низкоступенчатом языке ассемблера NASM.