

Отчёт по лабораторной работе №4

Архитектура вычислительных систем

Кроитору Екатерина

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выполнение самостоятельной работы	11
	Список литературы	13

Список иллюстраций

4.1 файл 10

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . .	7
-----	---	---

1 Цель работы

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

2 Задание

1. В соответствующем каталоге сделайте отчёт по лабораторной работе No4 в формате Markdown. В качестве отчёта необходимо предоставить отчёты в 3 форматах: pdf, docx и md.
2. Загрузите файлы на github.

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/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 об Unix см. в [1–6].

4 Выполнение лабораторной работы

- 1) Создаём каталог для работы с программами на языке ассемблера NASM (рис. ??)

```
ekroitoru@dk3n37 ~ $ cd work
ekroitoru@dk3n37 ~/work $ mkdir arch-pc
ekroitoru@dk3n37 ~/work $ cd arch-pc
ekroitoru@dk3n37 ~/work/arch-pc $ mkdir lab04
ekroitoru@dk3n37 ~/work/arch-pc $ cd lab04
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $
```

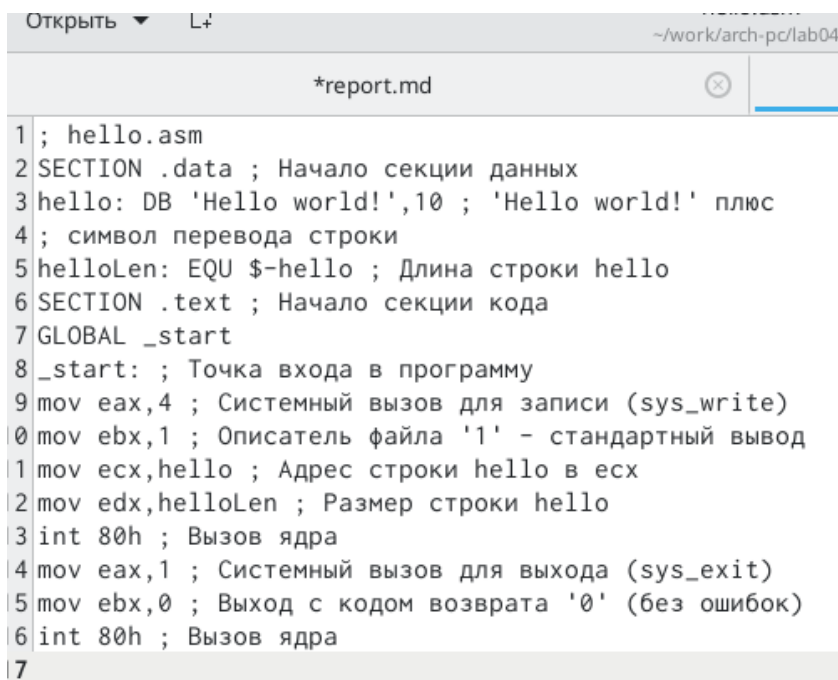
- 2) Создаём текстовый

файл с именем hello.asm и открываем этот файл с помощью любого текстового редактора gedit: (рис. ??)

```
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $ touch hello.asm
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $ gedit hello.asm
bash: gedit: команда не найдена
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $ gedit hello.asm
```

- 3) водим в него следу-

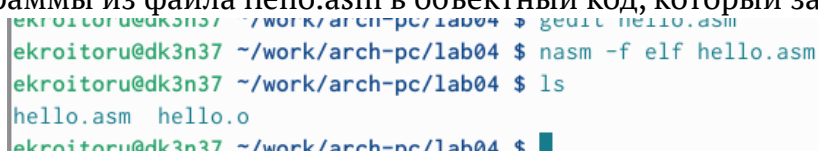
ющий текст: (рис. ??)



```
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
17
```

4) NASM превращает

текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать следующее (рис. ??) Т. к. текст программы набран без ошибок, транслятор преобразует текст программы из файла hello.asm в объектный код, который записан в файл hello.o.



```
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $ gcc hello.asm
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $ ls
hello.asm hello.o
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $
```

5) С помощью коман-

ды ls проверим, что объектный файл был создан. У нас есть два файла hello.asm и hello.o. Следующая команда скомпилирует исходный файл hello.asm в obj.o , при этом формат выходного файла будет elf, и в него будут включены символы для отладки (опция -g), кроме того, создается файл листинга list.lst .Выполним следующую команду: (рис. ??)



```
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hell
o.asm
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $ ls
hello.asm hello.o list.lst obj.o
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $
```

6) Чтобы получить ис-

полняемую программу, объектный файл необходимо пе- редать на обработку компоновщику, а потом с командой ls проверим содержи- мое: (рис. ??)

```

ekroitoru@dk3n37 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $

```

7) Ключ -o с последующим значением задаст в данном случае имя создаваемого исполняемого файла. Выполним следующую команду Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику, а потом с командой ls проверим содержимое: (рис. ??)

```

ekroitoru@dk3n37 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
ld: предупреждение: невозможно найти символ входа _start; начальный адрес не установлен
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $

```

8) Запустим на выполнение созданный исполняемый файл, находящийся в текущем каталоге, набрав в командной строке ./hello: (рис. 4.1)

```

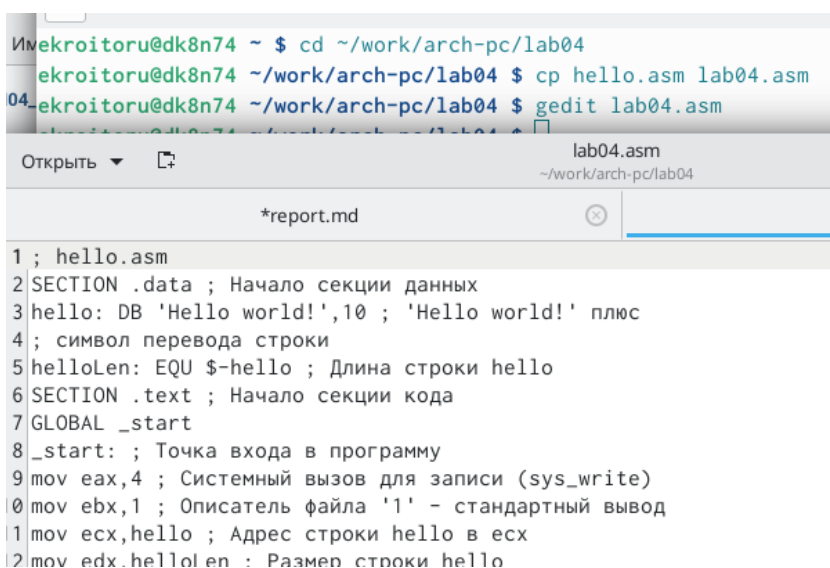
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $ ./hello
Hello world!
ekroitoru@dk3n37 ~/work/arch-pc/lab04 $

```

Рис. 4.1: файл

5 Выполнение самостоятельной работы

- 1) В каталоге ~/work/arch-pc/lab04 с помощью команды `cp` создали копию файла `hello.asm` с именем `lab04.asm`. (рис. ??)



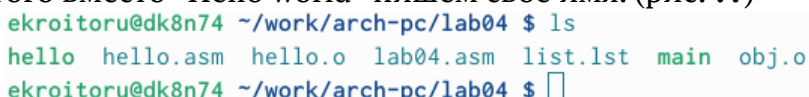
The screenshot shows a terminal window with the following commands and output:

```
ekroitoru@dk8n74 ~ $ cd ~/work/arch-pc/lab04
ekroitoru@dk8n74 ~/work/arch-pc/lab04 $ cp hello.asm lab04.asm
ekroitoru@dk8n74 ~/work/arch-pc/lab04 $ gedit lab04.asm
```

Below the terminal, the text editor shows the content of `lab04.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
```

- 2) С помощью текстового редактора вносим изменения в текст программы в файле `lab04.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с фамилией и именем. Для этого вместо “`Hello world`” пишем своё имя. (рис. ??)



The screenshot shows a terminal window with the following commands and output:

```
ekroitoru@dk8n74 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o lab04.asm list.lst main obj.o
ekroitoru@dk8n74 ~/work/arch-pc/lab04 $
```

- 3) Проводим схожие действия с лабораторной работой, но изменяем название файлов. (рис. ??)

4) Оттранслируем по-

5) Переносим файлы

6) Загружаем файлы

```

kroitoru@dk8n74 ~/work/study/2022-2023/Архитектура/2022-2023/2022-2023/5/report $ git push
Перечисление объектов: 26, готово.
Подсчет объектов: 100% (26/26), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (20/20), готово.
Запись объектов: 100% (20/20), 409.39 КиБ | 3.05 МБ/с
Всего 20 (изменений 3), повторно использовано 0 (объектов 0)

```

Я освоил процедуру компиляции и сборки программ, написанных на ассемблере NASM.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: <https://www.gnu.org/software/bash/manual/>.
 2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
 3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
 4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
 5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
 6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
-
1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: <https://www.gnu.org/software/bash/manual/>.
 2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
 3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
 4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
 5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
 6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.