

Отчёт по лабораторной работе №4

дисциплина: Архитектура компьютера

Бражко Александра Александровна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выполнение самостоятельной работы	11
6	Выводы	13
	Список литературы	14

Список иллюстраций

4.1	Создание каталога	8
4.2	переход в каталог	8
4.3	Создание и открытие текстового файла hello.asm с помощью gedit .	8
4.4	Ввод текста	9
4.5	Выполнение программ	9
4.6	Передача файла компоновщику	9
4.7	Проверка	9
4.8	Запуск файла	10
5.1	Копирование и проверка	11
5.2	Корректировка кода	11
5.3	Скопированные файлы	12

Список таблиц

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

1 Цель работы

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

2 Задание

1. Заполнение отчета по выполнению лабораторной работы №4 с помощью языка разметки Markdown.
2. Задание для самостоятельной работы.

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 про Unix см. в [1–4].

4 Выполнение лабораторной работы

Создаём каталог для работы с программами на языке ассемблера NASM (рис. 4.1).

```
aabrazhko@dk2n25 ~ $ mkdir -p ~/work/arch-pc/lab04
```

Рис. 4.1: Создание каталога

Переходим в созданный каталог (рис. 4.2).

```
aabrazhko@dk2n25 ~ $ cd ~/work/arch-pc/lab04
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $
```

Рис. 4.2: переход в каталог

Создаём текстовый файл с именем hello.asm и открываем этот файл с помощью любого текстового редактора, например, gedit (рис. 4.3).

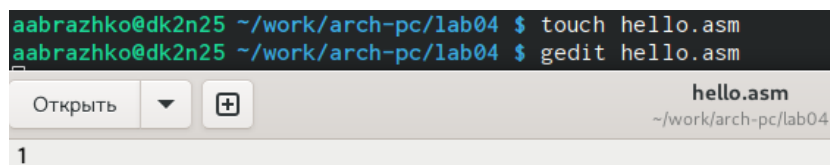
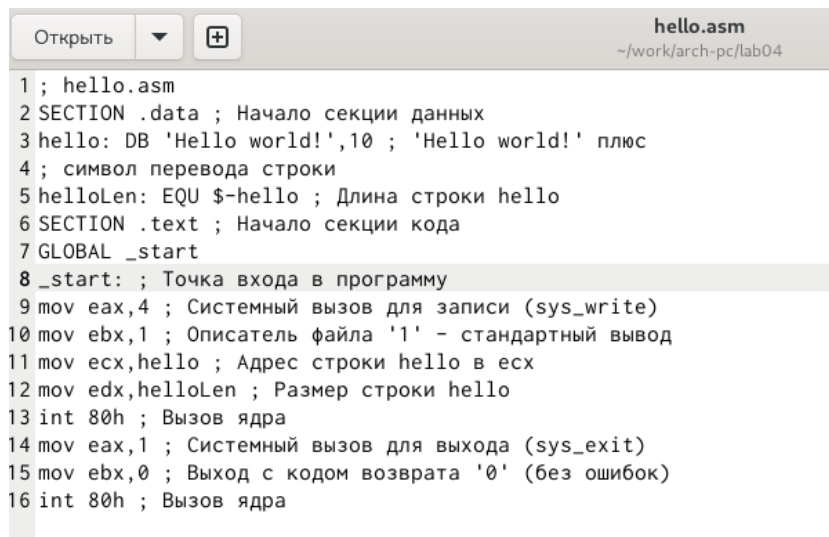


Рис. 4.3: Создание и открытие текстового файла hello.asm с помощью gedit

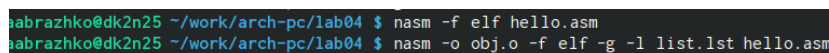
Вводим текст, предоставленный в файле для лабораторной работы №4 (рис. 4.4).



```
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
```

Рис. 4.4: Ввод текста

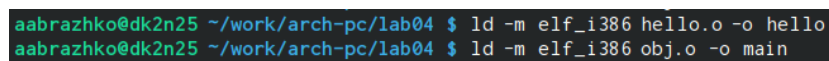
Чтобы скомпилировать написанный выше текст программы Hello world! выполним несколько программ (рис. 4.5).



```
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Рис. 4.5: Выполнение программ

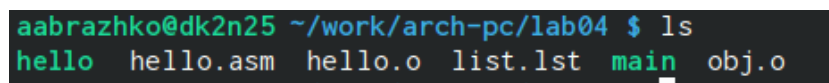
Передаём объектный файл на обработку компоновщику, чтобы получить исполняемую программу (рис. 4.6).



```
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
```

Рис. 4.6: Передача файла компоновщику

Проверка работы команд (рис. 4.7).



```
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o list.lst main obj.o
```

Рис. 4.7: Проверка

Запускаем на выполнение созданный исполняемый файл, находящийся в текущем каталоге (рис. 4.8).

```
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $ ./hello  
Hello world!  
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $
```

Рис. 4.8: Запуск файла

5 Выполнение самостоятельной работы

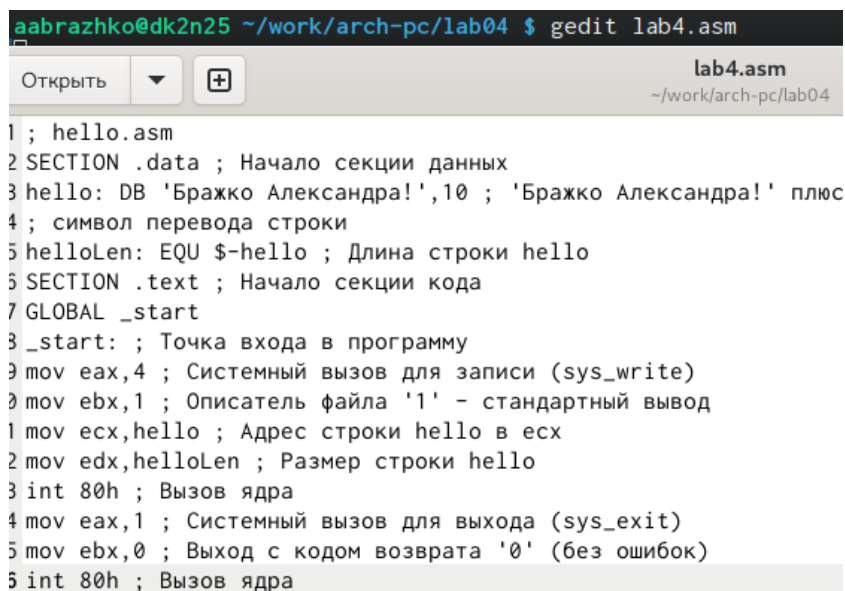
В каталоге `~/work/arch-pc/lab04` создаём копию файла `hello.asm` с именем `lab4.asm` и проводим проверку (рис. 5.1).

```
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
```

Рис. 5.1: Копирование и проверка

С помощью текстового редактора в файле `lab4.asm` вместо `Hello world!` вводим свои ФИ, чтобы они вывелись (рис. 5.2).

```
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $ gedit lab4.asm
```



```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Бражко Александра!',10 ; 'Бражко Александра!' 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
```

Рис. 5.2: Корректировка кода

Оттранслируем полученный текст программы lab4.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. (рис. ??, рис. ??).

```
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst lab4.asm
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab4
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $ ./lab4
```

```
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $
```

Бражко Александра!

```
aabrazhko@dk2n25 ~/work/arch-pc/lab04 $
```

Копируем файлы в локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/ (рис. 5.3).

sanyadruzko@gmail.com

Name

..

bib

image

pandoc

Makefile

hello.asm

lab4.asm

Лаб.4_Бражко_НММ6д-01-24.md

При скатии изменений используется до 6 потоков
Сжатие объектов: 100% (10/10), готово.
Запись объектов: 100% (10/10), 1.95 КиБ | 1.95 МиБ/с, готово.
Total 10 (delta 7), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (7/7), completed with 6 local objects.
To github.com:AlexandraBrazhko/study_2024-2025_arch-pc.git
32725f1..4cfea87 master -> master

```
aabrazhko@dk4n65 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04/report $ ls
bib hello.asm image lab4.asm Makefile pandoc Лаб.4_Бражко_НММ6д-01-24.md
aabrazhko@dk4n65 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04/report $
```

feat(main): make structure course and add reports	2 weeks ago
add labs/lab04/report hello.asm and lab4.asm	54 minutes ago
add labs/lab04/report hello.asm and lab4.asm	54 minutes ago
-	1 minute ago

Рис. 5.3: Скопированные файлы

6 Выводы

В результате выполнения данной лабораторной работы я освоила процедуры оформления отчетов с помощью легковесного языка разметки Markdown.

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.