

Отчет по лабораторной работе №4

Дисциплина: архитектура компьютера

Козина Дарья Александровна

Содержание

Цель работы	4
Задание.....	5
Выполнение лабораторной работы.....	6
Выполнение заданий для самостоятельной работы	9
Выводы	11

Список иллюстраций

Каталог для работы	6
Создание файла	6
Открытие файла в текстовом редакторе	6
Заполнение файла.....	7
Компиляция текста программы.....	7
Компиляция текста программы.....	7
Передача файла на обработку.....	7
Передача файла на обработку.....	8
Запуск файла.....	8
Копирование файла.....	9
Редактирование программы.....	9
Работа с файлом.....	9
Запуск файла.....	9
Копирование файлов.....	10
Загрузка файлов на GitHub	10

Цель работы

Освоить написанные на ассемблере NASM процедуры компиляции и сборки программ.

Задание

1. Создать программу Hello word!;
2. Поработать с транслятором NASM;
3. Поработать с расширенным синтаксисом командной строки NASM;
4. Поработать с компоновщиком LD;
5. Запустить исполняемый файл;
6. Выполнить задание для самостоятельной работы.

Выполнение лабораторной работы

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

```
dakozina@vbox:~$ mkdir -p ~/work/arch-pc/lab04
dakozina@vbox:~$ cd ~/work/arch-pc/lab04
dakozina@vbox:~/work/arch-pc/lab04$
```

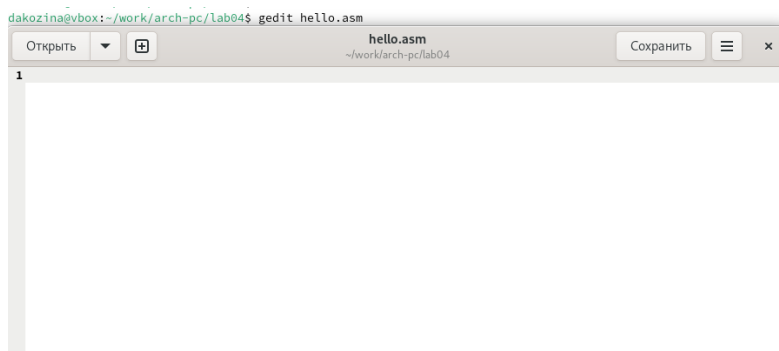
Каталог для работы

Создадим текстовый файл с именем hello.asm (рис. [-@fig:002]).

```
dakozina@vbox:~/work/arch-pc/lab04$ touch hello.asm
dakozina@vbox:~/work/arch-pc/lab04$
```

Создание файла

Откроем созданный файл с помощью текстового редактора gedit (рис. [-@fig:003]).



Открытие файла в текстовом редакторе

Заполняем файл, вставляя в него программу для вывода “Hello word!” (рис. [-@fig:004]).

```
Открыть *hello.asm
~/work/arch-pc/lab04
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
7 SECTION .text ; Начало секции кода
8     GLOBAL _start
9
10 _start: ; Точка входа в программу
11     mov eax,4 ; Системный вызов для записи (sys_write)
12     mov ebx,1 ; Описатель файла '1' - стандартный вывод
13     mov ecx,hello ; Адрес строки hello в ecx
14     mov edx,helloLen ; Размер строки hello
15     int 80h ; Вызов ядра
16
17     mov eax,1 ; Системный вызов для выхода (sys_exit)
18     mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
19     int 80h ; Вызов ядра
```

Заполнение файла

С помощью транслятора NASM превращаем текст программы в объективный код. Командой ls проверяем выполнение команды nasm (рис. [-@fig:005]).

```
dakozina@vbox:~/work/arch-pc/lab04$ nasm -f elf hello.asm
dakozina@vbox:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
dakozina@vbox:~/work/arch-pc/lab04$
```

Компиляция текста программы

С помощью команды скомпилируем файл, включим в него символы для отладки, создадим файл листинга. Командой ls проверяю правильность выполнения (рис. [-@fig:006]).

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

Компиляция текста программы

Передаем объектный файл на обработку компоновщику. Командой ls проверяем создание файла (рис. [-@fig:007]).

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

Передача файла на обработку

Выполняем следующую команду (рис. [-@fig:008]). Исполняемый файл будет иметь имя main, так как после ключа -o было задано значение main. Объектный файл, из которого собран этот исполняемый файл, имеет имя obj.o

```
dakozina@vbox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o mai
dakozina@vbox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst main obj.o
```

Передача файла на обработку

Запускаем созданный исполняемый файл на выполнение (рис. [-@fig:009]).

```
dakozina@vbox:~/work/arch-pc/lab04$ ./hello
Hello world!
```

Запуск файла

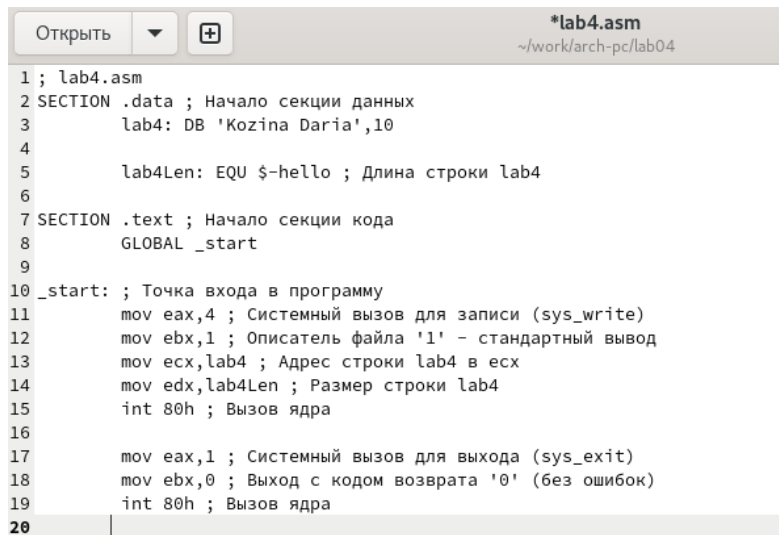
Выполнение заданий для самостоятельной работы

С помощью команды `cp` создадим копию файла `hello.asm` с новым именем (рис. [-@fig:010]).

```
dakozina@vbox:~$ cd ~/work/arch-pc/lab04
dakozina@vbox:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
dakozina@vbox:~/work/arch-pc/lab04$
```

Копирование файла

С помощью текстового редактора открываем файл, редактируем программу, чтобы она выводила мои фамилию и имя (рис. [-@fig:011]).



Редактирование программы

Оттранслируем полученный файл в объектный файл, выполним компоновку объектного файла, командой `ls` проверяем выполнение команд (рис. [-@fig:012]).

```
dakozina@vbox:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
dakozina@vbox:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list.lst  main  obj.o
dakozina@vbox:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
dakozina@vbox:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
```

Работа с файлом

Запустим полученный файл(рис. [-@fig:013]).

```
dakozina@vbox:~/work/arch-pc/lab04$ ./lab4
Kozina Daria
```

Запуск файла

Скопируем файлы `hello.asm` и `lab4.asm` в локальный репозиторий в каталог курса, командой `ls` проверим выполнение команды (рис. [-@fig:014]).

```
dakozina@vbox:~/work/arch-pc/lab04$ cp hello.asm ../../study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04
dakozina@vbox:~/work/arch-pc/lab04$ cp lab4.asm ../../study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04
dakozina@vbox:~/work/arch-pc/lab04$ cd ../../study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04
dakozina@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm lab4.asm presentation report
```

Копирование файлов

Загрузим файлы на GitHub (рис. [-@fig:015]).

```
dakozina@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
dakozina@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -m 'feat(main):add files lab-4'
[master 1b4e95c] feat(main):add files lab-4
6 files changed, 297 insertions(+)
create mode 100644 labs/lab03/report/L03_Kozina_otchet.docx
create mode 100644 labs/lab03/report/L03_Kozina_otchet.pdf
create mode 100644 labs/lab03/report/L03_Kozina_otchet (Копия).pdf
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
create mode 100644 labs/lab04/report/L04_Kozina_otchet.md
dakozina@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 24, готово.
Подсчет объектов: 100% (24/24), готово.
Сжатие объектов: 100% (18/18), готово.
Запись объектов: 100% (18/18), 1.35 Миб | 714.00 Киб/с, готово.
Total 18 (delta 9), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (9/9), completed with 4 local objects.
To github.com:dakozina/study_2024-2025_arh-pc.git
  22d36dd..1b4e95c master -> master
```

Загрузка файлов на GitHub

Выводы

В ходе лабораторной работы мы освоили написанные на ассемблере NASM процедуры компиляции и сборки программ.