

Отчёт по лабораторной работе №4

Создание и процесс обработки программ на языке ассемблера NASM

Алексей Павлович Корчагин

Содержание

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

Список иллюстраций

2.1	Схема ЭВМ	6
3.1	Создал каталог	8
3.2	Перемещение в каталог	8
3.3	Создание файла	8
3.4	Скопированный текст	9
3.5	Превращение файла	9
3.6	Файлы hello.asm и hallo.o	9
3.7	Компиляция файла	9
3.8	файлы обработанные компоновщиком	9
3.9	Передача файла компоновщику	10
3.10	Выполнение файла	10
4.1	Файл lab04	11
4.2	Изменения в файле	11
4.3	Тарнсляция текста	11
4.4	Компоновка файла	11
4.5	Выполнение файла	12
4.6	Копирование файла	12

Список таблиц

1 Цель работы

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

2 Теоретическое введение

Основными функциональными элементами любой электронно-вычислительной машины (ЭВМ) являются центральный процессор, память и периферийные устройства. В состав центрального процессора входят: арифметико-логическое устройство, устройство управления, регистры.

На иллюстрации ниже приведена структурная схема ЭВМ (рис. 2.1).

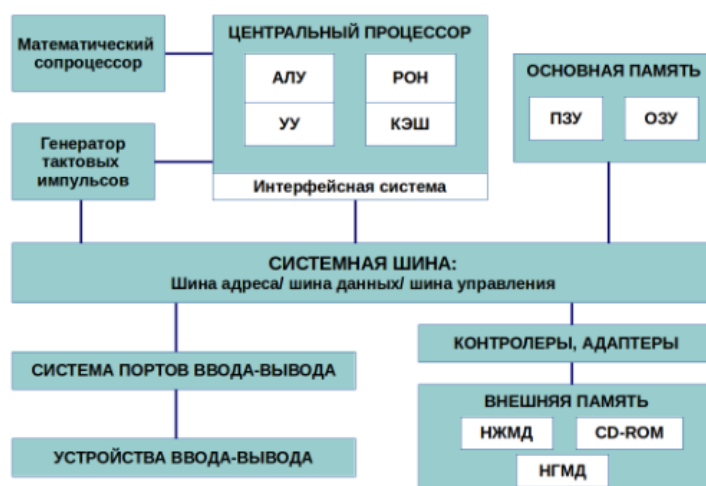


Рис. 2.1: Схема ЭВМ

Программы, которые написаны на языке ассемблера, не уступают в качестве и превосходят по скорости, программам, написанным на машинном языке, так как транслятор просто переводит мнемонические обозначения команд в числовые последовательности бит (нулей и единиц). Используемые мнемоники обычно одинаковы для всех процессоров одной архитектуры или семейства архи-

тектур (среди широко известных — мнемоники процессоров и контроллеров x86, ARM, SPARC, PowerPC, M68k). Таким образом для каждой архитектуры существует свой ассемблер и свой язык ассемблера. Наиболее распространёнными ассемблерами для архитектуры x86 являются: • для DOS/Windows: Borland Turbo Assembler (TASM), Microsoft Macro Assembler (MASM) и Watcom assembler (WASM); • для GNU/Linux: gas (GNU Assembler), использующий AT&T-синтаксис, в отличие от большинства других популярных ассемблеров, которые используют Intel-синтаксис.

3 Выполнение лабораторной работы

Создаю каталог lab04 (рис. 3.1).

```
apkorchagin@dk2n24 ~ $ mkdir -p ~/work/arch-pc/lab04
apkorchagin@dk2n24 ~ $
```

Рис. 3.1: Создал каталог

Перехожу в каталог lab04(рис. 3.2).

```
apkorchagin@dk2n24 ~ $ cd ~/work/arch-pc/lab04
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $
```

Рис. 3.2: Перемещение в каталог

Создаю текстовый файл с именем hello.asm(рис. 3.3).

```
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ touch hello.asm
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $
```

Рис. 3.3: Создание файла

открываю файл hello.asm и копирую текст в файл hello.asm(рис. 3.4).


```

1; hello.asm
2SECTION .data ; Начало секции данных
3hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4; символ перевода строки
5helloLen: EQU $-hello ; Длина строки hello
6SECTION .text ; Начало секции кода
7GLOBAL _start
8_start: ; Точка входа в программу
9mov eax,4 ; Системный вызов для записи (sys_write)
10mov ebx,1 ; Описатель файла '1' - стандартный вывод
11mov ecx,hello ; Адрес строки hello в ecx
12mov edx,helloLen ; Размер строки hello
13int 80h ; Вызов ядра
14mov eax,1 ; Системный вызов для выхода (sys_exit)
15mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
16int 80h ; Вызов ядра

```

Рис. 3.4: Скопированный текст

Превращаю файл hello.asm в объектный код, объектный файл называется hello.o (рис. 3.5).

```

apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $

```

Рис. 3.5: Превращение файла

Проверяю корректность выполнения предыдущей команды (рис. 3.6).

```

apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ ls
hello.asm hello.o
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $

```

Рис. 3.6: Файлы hello.asm и hallo.o

Компилирую файл hello.asm в obj.o и создаю файл list.lst (рис. 3.7).

```

apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ ls
hello.asm hello.o list.lst obj.o
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $

```

Рис. 3.7: Компиляция файла

Передаю файл hello.o на обработку компоновщику(рис. 3.8).

файлы обработанные компоновщиком

Рис. 3.8: файлы обработанные компоновщиком

Передаю файл `obj.o` на обработку компоновщику, исполняемый файл будет называться `main`, а объектный файл будет называться `obj.o` (рис. ??).

```
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $
```

Рис. 3.9: Передача файла компоновщику

Запускаю на выполнение созданный исполняемый файл (рис. 3.10).

```
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ ./hello
Hello world!
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $
```

Рис. 3.10: Выполнение файла

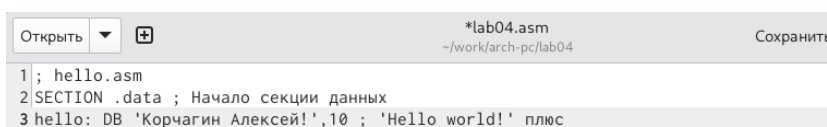
4 Задания самостоятельной работы

Создаю файл с новым названием (рис. 4.1).

```
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ cp hello.asm lab04.asm
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab04.asm  list.lst  main  obj.o
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $
```

Рис. 4.1: Файл lab04

Вношу изменения текста в файл lab04(рис. 4.2).



```
Открыть  +  *lab04.asm  Сохранить
~/work/arch-pc/lab04
1; hello.asm
2SECTION .data ; Начало секции данных
3hello: DB 'Корчагин Алексей!',10 ; 'Hello world!' плюс
```

Рис. 4.2: Изменения в файле

Транслирую текст в объектный файл (рис. 4.3).

```
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ gedit lab04.asm
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ nasm -o lab04.o -f elf -g -l list.lst lab04.asm
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab04.asm  lab04.o  list.lst  main  obj.o
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $
```

Рис. 4.3: Тарнсляция текста

Выполняю компоновку объектного файла (рис. 4.4).

```
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab04.o -o lab04
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab04  lab04.asm  lab04.o  list.lst  main  obj.o
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $
```

Рис. 4.4: Компоновка файла

Запускаю файл (рис. 4.5).

```
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ ./lab04
Корчагин Алексей!
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $
```

Рис. 4.5: Выполнение файла

Копирую файл в другой каталог(рис. 4.6).

```
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ cp hello.asm ~/work/study/2023-2024/"Архитектура
компьютера"/arch-pc/labs/lab04
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $ cp lab04.asm ~/work/study/2023-2024/"Архитектура
компьютера"/arch-pc/labs/lab04
apkorchagin@dk2n24 ~/work/arch-pc/lab04 $
```

Рис. 4.6: Копирование файла

5 Выводы

В ходе выполнения лабораторной работы я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.

Список литературы