

Отчет по лабораторной работе №4

Архитектура компьютера

Сафиуллина Айлина Саяровна

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 5 |
| 2 | Теоретическое введение | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 4 | Выводы | 10 |

Список иллюстраций

| | | |
|-----|---|---|
| 3.1 | Создание каталога и файла | 7 |
| 3.2 | текст листинга | 7 |
| 3.3 | Трансляция, линковка и запуск программы | 8 |
| 3.4 | Измененный текст программы | 9 |
| 3.5 | Результат программы | 9 |

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. Взаимодействие этих устройств осуществляется через общую шину, к которой они подключены. Физически шина представляет собой большое количество проводников, соединяющих устройства друг с другом. В современных компьютерах проводники выполнены в виде электропроводящих дорожек на материнской (системной) плате. Основной задачей процессора является обработка информации, а также организация координации всех узлов компьютера. В состав центрального процессора (ЦП) входят следующие устройства: • арифметико-логическое устройство (АЛУ) — выполняет логические и арифметические действия, необходимые для обработки информации, хранящейся в памяти; • устройство управления (УУ) — обеспечивает управление и контроль всех устройств компьютера; • регистры — сверхбыстрая оперативная память небольшого объёма, входящая в состав процессора, для временного хранения промежуточных результатов выполнения инструкций; регистры процессора делятся на два типа: регистры общего назначения и специальные регистры.

3 Выполнение лабораторной работы

С помощью команды `mkdir` я создала каталог `lab04`, затем перешла в него с помощью команды `cd` и создала файл `hello.asm`. (рис. 3.1).

```
assafiullina@dk1n22 ~ $ mkdir -p ~/work/arch-pc/lab04
assafiullina@dk1n22 ~ $ cd ~/work/arch-pc/lab04
assafiullina@dk1n22 ~/work/arch-pc/lab04 $ touch hello.asm
assafiullina@dk1n22 ~/work/arch-pc/lab04 $
```

Рис. 3.1: Создание каталога и файла

С помощью команды `ms` открыла созданный файл и внесла в него текст листинга из лабораторной работы. (рис. 3.2).

```
hello.asm      [----] 20 L:[ 1+15 16/ 16] *(810 / 810b) <EOF>      [*][X]
hello.asm
SECTION .data ; Начало секции данных
hello: DB 'Hello world!',10 ; 'Hello world!' плюс
; символ перевода строки
helloLen: EQU $-hello ; Длина строки hello
SECTION .text ; Начало секции кода
GLOBAL _start
_start: ; Точка входа в программу
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,hello ; Адрес строки hello в ecx
mov edx,helloLen ; Размер строки hello
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
int 80h ; Вызов ядра
```

Рис. 3.2: текст листинга

При помощи команды `nasm` я выполнила трансляцию файла. В результате был создан объектный файл `hello.o`. После повторно выполнила трансляцию с использованием дополнительных опций команды `nasm`. В результате этой операции были созданы файл листинга `list.lst`, объектный файл `obj.o`, и в программу была добавлена отладочная информация. При помощи команды `ld` я выполнила линковку и получила исполняемый файл. Повторно выполнила линковку для объектного файла `obj.o` и получила исполняемый файл с именем `main`. После этого запустила оба исполняемых файла и проверила их работу.(рис. 3.3).

```
assafiullina@dk1n22 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
assafiullina@dk1n22 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst h
ello.asm
assafiullina@dk1n22 ~/work/arch-pc/lab04 $ ls
hello.asm hello.o list.lst obj.o
assafiullina@dk1n22 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
assafiullina@dk1n22 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
assafiullina@dk1n22 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o list.lst main obj.o
assafiullina@dk1n22 ~/work/arch-pc/lab04 $ ./hello
Hello world!
assafiullina@dk1n22 ~/work/arch-pc/lab04 $
```

Рис. 3.3: Трансляция, линковка и запуск программы

Для выполнения задания я изменила текст сообщения “Hello world” на свое имя (рис. 3.4).


```

hello.asm      [-M--] 17 L:[ 1+ 1  2/ 15] *(72 / 793b) 0039 0x027 [*][X]
SECTION .data ; Начало секции данных
hello: DB 'Ailina',10 ; 'Hello world!' плюс
; символ перевода строки
helloLen: EQU $-hello ; Длина строки hello
SECTION .text ; Начало секции кода
GLOBAL _start
_start: ; Точка входа в программу
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,hello ; Адрес строки hello в ecx
mov edx,helloLen ; Размер строки hello
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
int 80h ; Вызов ядра

```

Рис. 3.4: Измененный текст программы

После изменения текста я снова запустила программу и получила корректный результат. (рис. 3.5).

```

assafiullina@dk1n22 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
assafiullina@dk1n22 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
assafiullina@dk1n22 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab4
assafiullina@dk1n22 ~/work/arch-pc/lab04 $ ./lab4
Ailina
assafiullina@dk1n22 ~/work/arch-pc/lab04 $ 

```

Рис. 3.5: Результат программы

4 Выводы

В ходе выполнения данной лабораторной работы я освоила процесс компиляции и сборки программ на ассемблере NASM, научилась трансляции, линковке и добавлению отладочной информации. А еще изменила код программы для вывода своего имени в соответствии с заданием для самостоятельной работы.