

Отчет по лабораторной работе №5.

Дисциплина: Архитектура компьютера.

Лобанова Полина Иннокентьевна

1 Цель работы

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

2 Выполнение лабораторной работы

1. Создадим каталог для работы с программами на языке ассемблера NASM и перейдем в него.

```
[pilobanova@fedora ~]$ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/lab05  
[pilobanova@fedora ~]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/lab05
```

Рис. 2.1: Создание каталога.

2. Создадим текстовый файл hello.asm.

```
[pilobanova@fedora lab05]$ touch hello.asm  
[pilobanova@fedora lab05]$ ls  
hello.asm
```

Рис. 2.2: Создание текстового файла.

3. Откроем этот файл с помощью текстового редактора gedit и заполним его.

```
[pilobanova@fedora lab05]$ gedit hello.asm
```

Рис. 2.3: Открытие файла.

```
; 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. 2.4: Заполнение файла.

4. Скомпилируем исходный файл в объектный файл.

```
[pilobanova@fedora lab05]$ nasm -f elf hello.asm
[pilobanova@fedora lab05]$ ls
hello.asm  hello.o
```

Рис. 2.5: Компиляция исходного файла.

5. Скомпилируем исходный файл в obj.o.

```
[pilobanova@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[pilobanova@fedora lab05]$ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис. 2.6: Компиляция исходного файла.

6. Выполним компоновку исходного файла.

```
[pilobanova@fedora lab05]$ ld -m elf_i386 hello.o -o hello
[pilobanova@fedora lab05]$ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Рис. 2.7: Компоновка исходного файла.

```
[pilobanova@fedora lab05]$ ld -m elf_i386 obj.o -o main
[pilobanova@fedora lab05]$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

Рис. 2.8: Компоновка исходного файла.

7. Запустим исполняемый файл.

```
[pilobanova@fedora lab05]$ ./hello
Hello world!
```

Рис. 2.9: Запущенная программа.

3 Выполнение самостоятельной работы

1. Скопируем файл hello.asm в файл lab5.asm в каталоге ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05.

```
[pilobanova@fedora lab05]$ cp hello.asm lab5.asm
[pilobanova@fedora lab05]$ ls
hello  hello.asm  hello.o  lab5.asm  list.lst  main  obj.o
```

Рис. 3.1: Копирование файлов.

2. Откроем файл с помощью текстового редактора gedit и заполним его.

```
[pilobanova@fedora lab05]$ gedit lab5.asm
```

Рис. 3.2: Открытие файла.

```
; hello.asm
SECTION .data ; Начало секции данных
hello: DB 'lobanova Polina',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. 3.3: Заполнение файла.

3. Оттранслируем полученный текст программы в объектный файл. Выполним компоновку объектного файла. Запустим получившийся исполняемый файл.

```
[pilobanova@fedora lab05]$ cp hello.asm lab5.asm
[pilobanova@fedora lab05]$ ls
hello hello.asm hello.o lab5.asm list.lst main obj.o
[pilobanova@fedora lab05]$ gedit lab5.asm
[pilobanova@fedora lab05]$ nasm -f elf lab5.asm
[pilobanova@fedora lab05]$ ls
hello hello.asm hello.o lab5.asm lab5.o list.lst main obj.o
[pilobanova@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst lab5.asm
[pilobanova@fedora lab05]$ ls
hello hello.asm hello.o lab5.asm lab5.o list.lst main obj.o
[pilobanova@fedora lab05]$ ld -m elf_i386 lab5.o -o lab5
[pilobanova@fedora lab05]$ ls
hello hello.asm hello.o lab5 lab5.asm lab5.o list.lst main obj.o
[pilobanova@fedora lab05]$ ld -m elf_i386 obj.o -o main
[pilobanova@fedora lab05]$ ls
hello hello.asm hello.o lab5 lab5.asm lab5.o list.lst main obj.o
[pilobanova@fedora lab05]$ ./lab5
Lobanova Polina
```

Рис. 3.4: Выполнение всех необходимых действий и запуск программы.

4. Скопируем файлы в локальный репозиторий и загрузим файлы на github.

```
[pilobanova@fedora lab05]$ cp hello.asm ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab05
[pilobanova@fedora lab05]$ cp lab5.asm ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab05
[pilobanova@fedora lab05]$ ls ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab05
hello.asm lab5.asm presentation report
[pilobanova@fedora lab05]$ git add .
[pilobanova@fedora lab05]$ git commit -am 'hello.asm lab5.asm'
[master b37c64e] hello.asm lab5.asm
39 files changed, 63 insertions(+), 2 deletions(-)
create mode 100755 lab05/hello
create mode 100644 lab05/hello.asm
create mode 100644 lab05/hello.o
create mode 100755 lab05/lab5
create mode 100644 lab05/lab5.asm
create mode 100644 lab05/lab5.o
create mode 100644 lab05/list.lst
create mode 100755 lab05/main
create mode 100644 lab05/obj.o
```

Рис. 3.5: Перенос файлов в локальный репозиторий и загрузка на github.

```
[pilobanova@fedora lab05]$ git push
Перечисление объектов: 29, готово.
Подсчет объектов: 100% (29/29), готово.
Сжатие объектов: 100% (20/20), готово.
Запись объектов: 100% (20/20), 3.60 КиБ | 1.80 МБ/с, готово.
Всего 20 (изменений 12), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (12/12), completed with 7 local objects.
To github.com:pilobanova/study_2022-2023_arh-pc.git
 e142ca7..b37c64e master -> master
[pilobanova@fedora lab05]$
```

Рис. 3.6: Загрузка файлов на github.

4 Выводы

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