

Отчет по лабораторной работе №5

Создание и процесс обработки программ на языке ассемблера NASM

Попова Елизавета Сергеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выполнения самостоятельной работы	10
5	Выводы	12

Список иллюстраций

3.1	Создание каталога lab05	7
3.2	Переход к каталогу lab05	7
3.3	Файл hello.asm	7
3.4	Содержание файла hello.asm	8
3.5	Компилирование кода	8
3.6	Компилирование файла	8
3.7	Компилирование файла	8
3.8	Работа файла hello	9
4.1	Файлы в каталоге lab05	10
4.2	Изменение кода в файле lab5.asm	10
4.3	Изменение файла	11
4.4	Запуск файла lab5	11
4.5	Загрузка на GitHub	11

Список таблиц

1 Цель работы

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

2 Задание

Написать простейший код на ассемблере NASM.

3 Выполнение лабораторной работы

- 1) Создаем каталог lab05 и переходим в него

```
[laisacreate@10 ~]$ mkdir ~/work/arch-pc/lab05
```

Рис. 3.1: Создание каталога lab05

```
[laisacreate@10 ~]$ cd ~/work/arch-pc/lab05
```

Рис. 3.2: Переход к каталогу lab05

- 2) Создаем файл hello.asm и открываем его при помощи команды gedit.

```
[laisacreate@10 lab05]$ touch hello.asm  
[laisacreate@10 lab05]$ gedit hello.asm
```

Рис. 3.3: Файл hello.asm

- 3) Вводим необходимый текст

```

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. 3.4: Содержание файла hello.asm

4) Компилируем код

```

[laisacrete@10 lab05]$ nasm -f elf hello.asm
[laisacrete@10 lab05]$ ls
hello.asm hello.o
[laisacrete@10 lab05]$

```

Рис. 3.5: Компилирование кода

5) Компиляция файла hello.asm в obj.o

```

[laisacrete@10 lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[laisacrete@10 lab05]$ ls
hello.asm hello.o list.lst obj.o

```

Рис. 3.6: Компилирование файла

6) Обработка компоновщиком файлов hello.o и obj.o

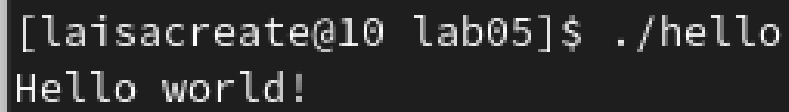
```

[laisacrete@10 lab05]$ ld -m elf_i386 hello.o -o hello
[laisacrete@10 lab05]$ ls
hello hello.asm hello.o list.lst obj.o
[laisacrete@10 lab05]$ ld -m elf_i386 obj.o -o main
[laisacrete@10 lab05]$ ls
hello hello.asm hello.o list.lst main obj.o
[laisacrete@10 lab05]$

```

Рис. 3.7: Компилирование файла

7) Запускаем файл hello

A terminal window with a dark background. The prompt is [laisacrete@10 lab05]\$ and the command entered is ./hello. The output is Hello world! on the next line.

```
[laisacrete@10 lab05]$ ./hello  
Hello world!
```

Рис. 3.8: Работа файла hello

4 Выполнения самостоятельной работы

- 1) При помощи команды `cp` делаем копию файла `hello.asm` с именем `lab5.asm`

```
[laisacrete@10 lab05]$ cp hello.asm lab5.asm
[laisacrete@10 lab05]$ ls
hello  hello.asm  hello.o  lab5.asm  list.lst  main  obj.o
```

Рис. 4.1: Файлы в каталоге lab05

- 2) При помощи редактора `gedit` вносим изменения в файл `lab5.asm`

```
1 ; hello.asm
2 SECTION .data                ; Начало секции данных
3     hello: DB 'Popova Elizaveta Sergeevna',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. 4.2: Изменение кода в файле lab5.asm

- 3) Оттранслируем полученный текст и выполним компоновку файла

```
[laisacrete@10 lab05]$ nasm -f elf lab5.asm
[laisacrete@10 lab05]$ ls
hello hello.asm hello.o lab5.asm lab5.o list.lst main obj.o
[laisacrete@10 lab05]$ nasm -o obj.o -f elf -g -l list.lst lab5.asm
[laisacrete@10 lab05]$ ls
hello hello.asm hello.o lab5.asm lab5.o list.lst main obj.o
[laisacrete@10 lab05]$ ld -m elf_i386 lab5.o -o lab5
[laisacrete@10 lab05]$ ls
hello hello.asm hello.o lab5 lab5.asm lab5.o list.lst main obj.o
[laisacrete@10 lab05]$ ld -m -elf_i386 lab5.o -o main
ld: не распознан режим эмуляции: -elf_i386
Поддерживаемые эмуляции: elf_x86_64 elf32_x86_64 elf_i386 elf_iarmcu elf_llor
f
[laisacrete@10 lab05]$ ld -m elf_i386 lab5.o -o main
[laisacrete@10 lab05]$ ls
hello hello.asm hello.o lab5 lab5.asm lab5.o list.lst main obj.o
```

Рис. 4.3: Изменение файла

4) Запускаем файл lab5

```
[laisacrete@10 lab05]$ ./lab5
Popova Elizaveta Sergeevna
```

Рис. 4.4: Запуск файла lab5

5) Скопируем файлы в локальный репозиторий и загрузим их на Github

```
[laisacrete@10 study_2022-2023_arh-pc]$ git add .
[laisacrete@10 study_2022-2023_arh-pc]$ git commit -am 'lab5'
[master 4519c40] lab5
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100755 labs/lab05/hello
create mode 100755 labs/lab05/lab5
[laisacrete@10 study_2022-2023_arh-pc]$ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 977 байтов | 977.00 КиБ/с, готово.
Всего 6 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:chistachill/study_2022-2023_arh-pc.git
63803b9..4519c40 master -> master
[laisacrete@10 study_2022-2023_arh-pc]$
```

Рис. 4.5: Загрузка на GitHub

5 Выводы

Мы освоили процедуры компиляции и сборки программы, написанные на ассемблере NASM