

Лабораторная работа №4

Создание и процесс обработки программ на языке ассемблера NASM

Турсунбоев Сардорбек

Содержание

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

Список иллюстраций

3.1	Рис. 1 Создание нового подкаталога	7
3.2	Рис. 2 Переход в созданный подкаталог	7
3.3	Рис. 3 Создание файла hello.asm	7
3.4	Рис. 4 Открою файл hello.asm в текстовом редакторе gedit	7
3.5	Рис. 5 Пример кода в текстовом редакторе	8
3.6	Рис. 6 Работа транслятора	8
3.7	Рис.7 Компиляция программы с расширенным синтаксисом	8
3.8	Рис. 8 Работа компоновщика ld	9
3.9	Рис.9 Передача файла obj.o компоновщику и проверка его работы	9
3.10	Рис. 10 Запуск программы hello	9
3.11	Рис. 11 Копирование файла hello.asm	9
3.12	Рис.12 Открытие файла lab4.asm	9
3.13	Рис.13 Код с внесёнными изменениями	10
3.14	Рис.14 Компиляция программы lab4.asm	10
3.15	Рис.15 Создание файла lab	10
3.16	Рис.16 Результат работы программы lab	11
3.17	Рис. 17 Использование утилиты cp для копирования программ в рабочий каталог	11

Список таблиц

1 Цель работы

Цель данной лабораторной работы – базовая ознакомление с языком NASM, а также освоить процедуры компиляции и сборки программ написанных на этом языке.

2 Задание

1. Создание программы Hello world на языке NASM
2. Работа с транслятором NASM
3. Работа с компоновщиком LD
4. Запуск исполняемого файла
5. Задания для самостоятельной работы

3 Выполнение лабораторной работы

Создам в рабочем каталоге work подкаталог в котором будет выполняться лабораторная работа(Рис.1).

```
[sktursunboev@fedora 2023-2024]$ mkdir -p ~/work/arch-pc/lab04
```

Рис. 3.1: Рис. 1 Создание нового подкаталога

Перейду в созданный нами подкаталог(Рис.2).

```
[sktursunboev@fedora 2023-2024]$ cd ~/work/arch-pc/lab04
```

Рис. 3.2: Рис. 2 Переход в созданный подкаталог

создам файл hello.asm утилитой touch(Рис.3).

```
[sktursunboev@fedora lab04]$ touch hello.asm
```

Рис. 3.3: Рис. 3 Создание файла hello.asm

Редактировать файл буду в текстовом редакторе gedit(Рис.4).

```
[sktursunboev@fedora lab04]$ gedit hello.asm
```

Рис. 3.4: Рис. 4 Открою файл hello.asm в текстовом редакторе gedit

Вставлю пример кода из лабораторной работы в текстовый редактор(Рис.5).

```

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. 3.5: Рис. 5 Пример кода в текстовом редакторе

Переведу текст программы hello.asm в объектный код с помощью транслятора(Рис.6).

```
[sktursunboev@fedora lab04]$ nasm -f elf hello.asm
```

Рис. 3.6: Рис. 6 Работа транслятора

Введу команду которая скомпилирует файл hello.asm в файл obj.o, так же использую ключом -g и с помощью ключа -l создам файл листинга list.lst(Рис.7).

```
[sktursunboev@fedora lab04]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Рис. 3.7: Рис.7 Компиляция программы с расширенным синтаксисом

Передаю файл hello.o компоновщику ld, задам имя создаваемого файла с помощью ключа -o(Рис.8).


```
[sktursunboev@fedora lab04]$ ld -m elf_i386 hello.o -o hello
```

Рис. 3.8: Рис. 8 Работа компоновщика ld

Передаю файл obj.o компоновщику. Файл будет иметь имя main, так как оно было указано после ключа -o(Рис.9).

```
[sktursunboev@fedora lab04]$ ld -m elf_i386 obj.o -o main
sktursunboev@fedora lab04]$ ls
hello hello.asm hello.o list.lst main obj.o
```

Рис. 3.9: Рис.9 Передача файла obj.o компоновщику и проверка его работы

Запущу сделанную нами программу hello(Рис.10).

```
[sktursunboev@fedora lab04]$ ./hello
Hello world!
```

Рис. 3.10: Рис. 10 Запуск программы hello

Утилитой cp создам копию hello.asm и назовём её lab4.asm. Проверим правильность копирования утилитой ls(Рис.11).

```
[sktursunboev@fedora lab04]$ cp hello.asm lab4.asm
[sktursunboev@fedora lab04]$ ls
hello hello.asm hello.o lab4.asm list.lst main obj.o
```

Рис. 3.11: Рис. 11 Копирование файла hello.asm

Открою файл lab4.asm в текстовом редакторе gedit(Рис.12).

```
[sktursunboev@fedora lab04]$ gedit lab4.asm
```

Рис. 3.12: Рис.12 Открытие файла lab4.asm

Вношу изменения в код так, чтобы программа lab4.asm выводила мои имя и фамилию(Рис.13).

```

SECTION .data ; Начало секции данных
hello: DB 'Турсунбоев Сарддробек',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. 3.13: Рис.13 Код с внесёнными изменениями

Скомпилирую получившийся код в объектный файл lab4.o(Рис.14).

```
[sktursunboev@fedora lab04]$ nasm -f elf lab4.asm
```

Рис. 3.14: Рис.14 Компиляция программы lab4.asm

Отправлю объектный код компоновщику для создания файла lab(Рис.15).

```
[sktursunboev@fedora lab04]$ ld -m elf_i385 lab4.o -o lab
```

Рис. 3.15: Рис.15 Создание файла lab

Запущу для проверки программу lab(Рис.16).

```
[sktursunboev@fedora lab04]$ ./lab
Турсунбоев Сардорбек
```

Рис. 3.16: Рис.16 Результат работы программы lab

Скопирую программы созданные в ходе этой лабораторной работы в рабочий каталог(Рис.17).

```
[sktursunboev@fedora lab04]$ cp hello.asm ~/work/study/2023-2024/Архитектура \ Компьютера/study_2023_2024_arh--pc/labs/lab04
[sktursunboev@fedora lab04]$ cp lab4.asm ~/work/study/2023-2024/Архитектура \ Компьютера/study_2023_2024_arh--pc/labs/lab04
```

Рис. 3.17: Рис. 17 Использование утилиты cp для копирования программ в рабочий каталог

4 Выводы

В ходе данной лабораторной работы я познакомился с языком NASM и научился работать с компилятором и компоновщиком