

Отчёт по лабораторной работе №4

Создание и процесс обработки на языке ассемблера NASM

Попутников Егор Сергеевич

Содержание

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

Список иллюстраций

3.1	Создание каталога	7
3.2	Создание файла hello.asm	7
3.3	Редактирование файла	8
3.4	Компиляция файла	8
3.5	Запуск исполняемого файла	9
4.1	Копирование файла	10
4.2	Редактирование файла	10
4.3	Копирование файлов в локальный репозиторий	10
4.4	Загрузка файлов на github	11
4.5	Загрузка файлов на github	11
4.6	Загрузка файлов на github	11

Список таблиц

1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

2 Задание

1. В каталоге `~/work/arch-рс/lab04` с помощью команды `ср` создайте копию файла `hello.asm` с именем `lab4.asm`
2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы `lab4.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы `hello.asm` и `lab4.asm` в Ваш локальный репозиторий в каталог `~/work/study/2023-2024/“Архитектура компьютера”/arch-рс/labs/lab04/`. Загрузите файлы на Github.

3 Выполнение лабораторной работы

Создадим каталог для работы с программами на языке ассемблера NASM.(3.1)

```
egor@espoputnikov-dk3n56:~$ mkdir -p ~/work/arch-pc/lab04  
egor@espoputnikov-dk3n56:~$ cd ~/work/arch-pc/lab04
```

Рис. 3.1: Создание каталога

Создадим текстовый файл с именем hello.asm и откроем его с помощью текстового редактора.(3.2)

```
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ touch hello.asm  
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ gedit hello.asm
```

Рис. 3.2: Создание файла hello.asm

Введём в него следующий текст:(3.3)

```

; 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

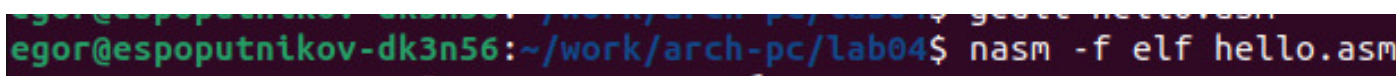
_start:                                       ; Точка входа в программу
    mov eax,4                               ; Системный вызов для записи (sys_write)
    mov ebx,1                               ; Описатель файла '1' - стандартный вывод
    mov ecx,hello                           ; Адрес строки 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. 3.3: Редактирование файла

Проведем компиляцию для приведённого выше файла:(3.4)



```

egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ nasm -f elf hello.asm

```

Рис. 3.4: Компиляция файла

Скомпилируем исходный файл в obj.o, объектный файл передадим на обработку компоновщику и запустим исполняемый файл:(3.5)


```
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ nasm -f elf hello.asm
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.a
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ ./hello
Hello world!
```

Рис. 3.5: Запуск исполняемого файла

4 Выполнение самостоятельной работы

В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создадим копию файла `hello.asm` с именем `lab4.asm`:(4.1)

```
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ cp ~/work/arch-pc/lab04/hello.asm ~/work/arch-pc/lab04/lab4.asm
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
```

Рис. 4.1: Копирование файла

С помощью команды `gedit` внесём изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с моей фамилией и моим именем:(4.2)

```
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ gedit lab4.asm
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ ./lab4
Попутников Егор
```

Рис. 4.2: Редактирование файла

Скопируем файлы `hello.asm` и `lab4.asm` в мой локальный репозиторий в каталог `~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/`:(4.3)

```
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ cp ~/work/arch-pc/lab04/hello.asm ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ cp ~/work/arch-pc/lab04/lab4.asm ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04
egor@espoputnikov-dk3n56:~/work/arch-pc/lab04$ cd ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04
egor@espoputnikov-dk3n56:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm  lab4.asm  presentation  report
```

Рис. 4.3: Копирование файлов в локальный репозиторий

Далее загрузим файлы на github:(4.4)(4.5)(4.6)

```
Архитектура компьютера/arch-pc$ git add .  
egor@espoputnikov-dk3n56:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 4.4: Загрузка файлов на github

```
egor@espoputnikov-dk3n56:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): add files ab-4'
```

Рис. 4.5: Загрузка файлов на github

```
egor@espoputnikov-dk3n56:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push  
Перечисление объектов: 15, готово.  
Подсчет объектов: 100% (15/15), готово.  
Сжатие объектов: 100% (10/10), готово.  
Запись объектов: 100% (10/10), 3.56 Миб | 2.89 Миб/с, готово.  
Всего 10 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0  
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
```

Рис. 4.6: Загрузка файлов на github

5 Выводы

Я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.