

Российский университет дружбы народов

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

Максимиано Герра

С.Б 1132215384

Москва, 2022

Введение:

В лабораторной работе, которая будет выполнена, вводится языковой ассемблер, который имеет много прибыли во время построения компьютера, программного обеспечения и т. д. Это ключевой инструмент для всех системных операций, поскольку он подключается через команды (которые обрабатываются и читаются как двоичные коды), компьютерное оборудование.

Лабораторная работа №5.pdf (página 10 de 18) — Modificado

```

и введите в него следующий текст:

; 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_wwrite)
    mov ebx,1 ; Описатель файла '1' - стандартный вывод
    mov ecx,hello ; Адрес строки hello в ех
    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

```

В отличие от многих современных высокоскоуровневых языков программирования, в ассемблерной программе каждая команда располагается на отдельной строке. Размещение нескольких команд на одной строке недопустимо. Синтаксис ассемблера NASM является чувствительным к регистру, т.е. есть разница между большими и малыми буквами.

5.3.2. Транслятор NASM

NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать:

78

Демидова А. В.

Справочник с соответствующей папкой был создан, также был создан документ, где получена информация, запрашиваемая в лаборатории

Лабораторная работа №5.pdf (página 11 de 18) — Modificado

между большими и малыми буквами.

5.3.2. Транслятор NASM

NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать:

78

Демидова А. В.

Архитектура ЭВМ

nasm -f elf hello.asm

Соответствующие команды были записаны для выполнения файла .asm, перейдите в файл .o

Лабораторная работа №5.pdf (página 11 de 18) — Modificado

5.3.2. Транслятор NASM

NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать:

```
nasm -f elf hello.asm
```

78 Демидова А. В.

Архитектура ЭВМ

Если текст программы набран без ошибок, то транслятор преобразует текст программы из файла hello.asm в объектный код, который записывается в файл hello.o. Таким образом, имена всех файлов получаются из имени входного файла и расширения по умолчанию. При наличии ошибок объектный файл не создается, а после запуска транслятора появятся сообщения об ошибках или предупреждениях.

С помощью команды ls проверьте, что объектный файл был создан. Какое имя имеет объектный файл?

NASM не запускают без параметров. Ключ -f указывает транслятору, что требуется создать бинарные файлы в формате ELF. Следует отметить, что формат elf64 позволяет создавать исполняемый код, работающий под 64-битными версиями Linux. Для 32-битных версий ОС указываем в качестве формата просто elf.

NASM всегда создает выходные файлы в текущем каталоге.

5.3.3. Расширенный синтаксис командной строки NASM

Полный вариант командной строки nasm выглядит следующим образом:

```
nasm [-@ косвенный_файл_настроек] [-o объектный_файл] [-f
↳ формат_объектного_файла] [-l листинг] [параметры...] [-]
```

Лабораторная работа №5.pdf (página 11 de 18) — Modificado

5.3.2. Транслятор NASM

NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать:

```
nasm -f elf hello.asm
```

78 Демидова А. В.

Архитектура ЭВМ

Если текст программы набран без ошибок, то транслятор преобразует текст программы из файла hello.asm в объектный код, который записывается в файл hello.o. Таким образом, имена всех файлов получаются из имени входного файла и расширения по умолчанию. При наличии ошибок объектный файл не создается, а после запуска транслятора появятся сообщения об ошибках или предупреждениях.

С помощью команды ls проверьте, что объектный файл был создан. Какое имя имеет объектный файл?

NASM не запускают без параметров. Ключ -f указывает транслятору, что требуется создать бинарные файлы в формате ELF. Следует отметить, что формат elf64 позволяет создавать исполняемый код, работающий под 64-битными версиями Linux. Для 32-битных версий ОС указываем в качестве формата просто elf.

NASM всегда создает выходные файлы в текущем каталоге.

5.3.3. Расширенный синтаксис командной строки NASM

Полный вариант командной строки nasm выглядит следующим образом:

```
nasm [-@ косвенный_файл_настроек] [-o объектный_файл] [-f
↳ формат_объектного_файла] [-l листинг] [параметры...] [-]
```

fedora [Работает]

```
[gagerra@fedora ~]$ nasm -f elf hello.asm
hello.asm: warning: label alone on a line without a colon might be in error [-wlabel-orphan]
[gagerra@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
hello.asm: warning: label alone on a line without a colon might be in error [-wlabel-orphan]
[gagerra@fedora lab05]$ ld -m elf_i386 hello.o -o hello
[gagerra@fedora lab05]$ ./hello
Hello world!
[gagerra@fedora lab05]$ ls
```

Arхитектура ЭВМ

Запуск исполняемого файла

Чтобы запустить на выполнение созданный исполняемый файл, нужно перейти в текущем каталоге, можно, набрав в командной строке ./hello

5.5. Задание для самостоятельной работы

- В каталоге ~/work/arch_pc/lab05 с помощью команды nasm скомпилируйте файл hello.asm с именем lab5.asm
- С помощью любого текстового редактора вставьте строку с вашими фамилией и именем в конец файла lab5.asm
- Оттранслируйте полученный текст программы. Выполните компоновку объектного файла. Укажите имя для исполняемого файла.
- Скопируйте файлы hello.asm и lab5.asm в каталог ~/work/study/2022-2023/архитектура_ЭВМ/лаб05/. Загрузите файлы на GitHub.

5.6. Содержание отчёта

Отчёт должен включать:

- Как видно из схемы на рисунке, что такое объектный файл и как его создавали.
- Команды для создания объектного файла.
- С помощью команды ls проконтролировать, что в каталоге не предполагается использовать следующие файлы:
 - o – для объектных файлов
 - без расширения – для исходных текстов
 - map – для файлов схемы
 - lib – для библиотек.
- Ключ -o для последующего назначения имени для объектного файла.
- Выполните следующую команду:

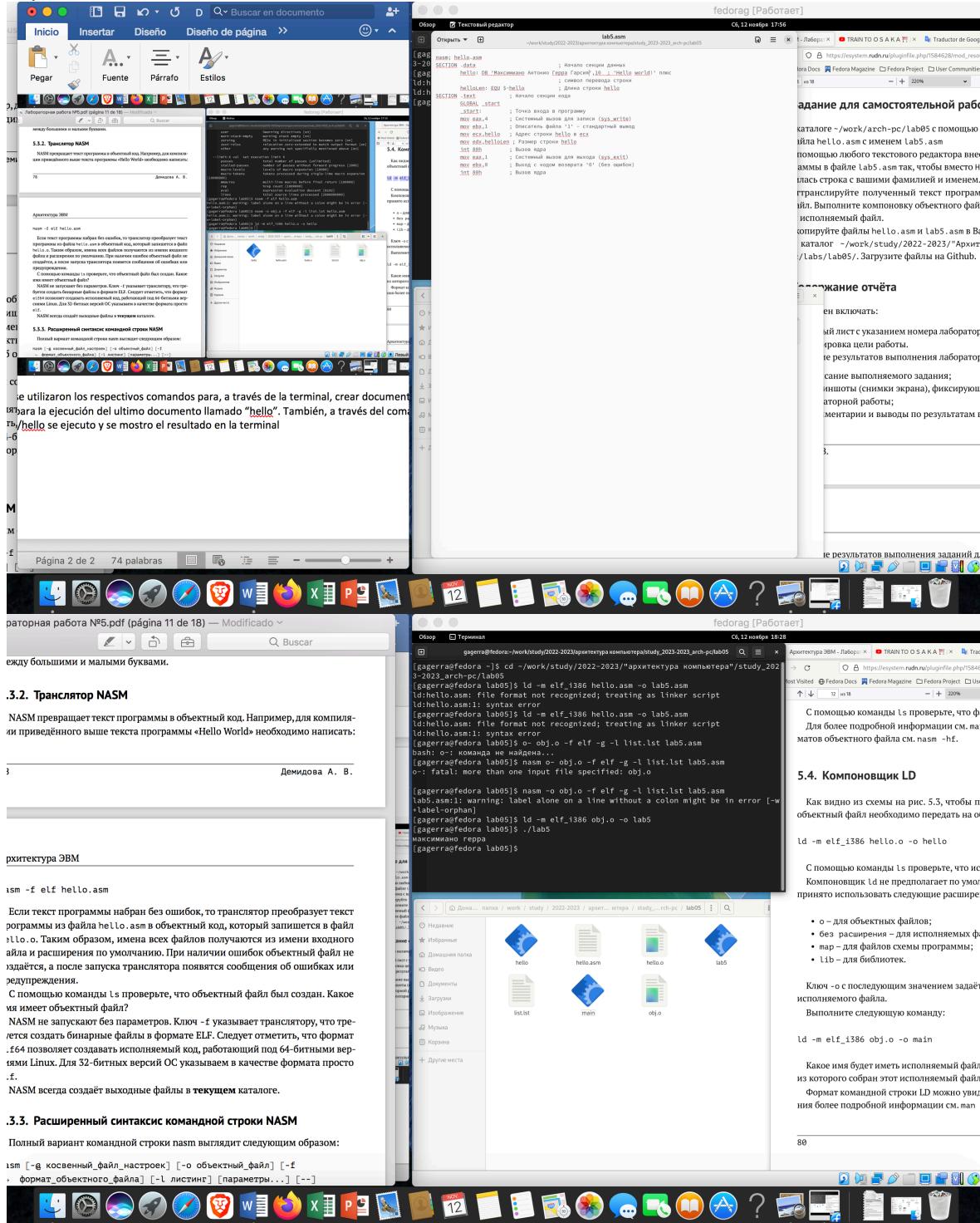
```
ld -m elf_i386 obj.o -o ma
```
- Какое имя будет иметь исполняемый файл?
- Формат командной строки и его более подробной информации.

80

Архитектура ЭВМ

Левый ⌂ + Левый ⌂

Соответствующие команды использовались, чтобы через терминал создал документы для выполнения последнего документа, называемого «Привет». Кроме того, через команду ./Hello она была выполнена, и результат был показан в терминале



3.2. Транслятор NASM

NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать:

3 Демидова А. В.

архитектура ЭВМ

asm -f elf hello.asm

Если текст программы набран без ошибок, то транслятор преобразует текст программы из файла hello.asm в объектный код, который записывается в файл hello.o. Таким образом, имена всех файлов получаются из имени входного файла и расширения по умолчанию. При наличии ошибок объектный файл не создаётся, а после запуска транслятора появятся сообщения об ошибках или редиректа.

С помощью команды ls проверьте, что объектный файл был создан. Какое имя имеет объектный файл?

NASM не запускают без параметров. Ключ -f указывает транслятору, что требуется создать бинарные файлы в формате ELF. Следует отметить, что формат .elf позволяет создавать исполняемый код, работающий под 64-битными версиями Linux. Для 32-битных версий ОС указываем в качестве формата просто .elf.

NASM всегда создаёт выходные файлы в текущем каталоге.

3.3. Расширенный синтаксис командной строки NASM

Полный вариант командной строки nasm выглядит следующим образом:

```
нasm [-o косвенный_файл_настроек] [-o объектный_файл] [-f
  формат_объектного_файла] [-l листинг] [параметры...] [-]
```

80

Каким образом можно использовать расширенный синтаксис командной строки?

Формат командной строки LD можно увидеть более подробной информацией см. man ld.

Ключ -o для объектных файлов;

• без расширения – для исполняемых файлов;

• map – для файлов схемы программы;

• lib – для библиотек.

Ключ -o – с последующим значением задаёт имя исполнимого файла.

Выполните следующую команду:

```
ld -m elf_i386 obj.o -o main
```

Какое имя будет иметь исполняемый файл? из которого собран этот исполняемый файл?

Формат командной строки LD можно увидеть более подробной информацией см. man ld.

В терминале и доске документов директоров были созданы формат и имя для выполнения LAB5, после этого его можно было выполнить, а имя и фамилия были напечатаны.

ВЫВОД:

Был сделан вывод, что функциональность языка ассемблера имеет большой потенциал, поскольку охватывает отличное поле компьютерного строительства