Создание и процесс обработки программ на языке ассемблера NASM

Лабораторная работа №4

Губайдуллина Софья Романовна

Содержание

1 Цель работы

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

2 Задание

Порядок выполнения лабораторной работы:

1) Программа Hello world!
2) Расширенный синтаксис командной строки NASM
3) Компоновщик LD
4) Запуск исполняемого файла
5) Задание для самостоятельной работы

3 Теоретическое введение

Язык ассемблера (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. Можно считать, что он больше любых других языков приближен к архитектуре ЭВМ и её аппаратным возможностям, что позволяет получить к ним более полный доступ, нежели в языках высокого уровня. Для того, чтобы писать программы на ассемблере, необходимо знать, какие регистры процессора существуют и как их можно использовать.

В основе вычислительного процесса ЭВМ лежит принцип программного управления. Это означает, что компьютер решает поставленную задачу как последовательность действий, записанных в виде программы. Программа состоит из машинных команд, которые указыва- ют, какие операции и над какими данными (или операндами), в какой последовательности необходимо выполнить. Следует отметить, что процессор понимает не команды ассемблера, а последовательности из нулей и единиц — машинные коды. Преобразование или трансляция команд с языка ассемблера в исполняемый машинный код осуществляется специальной программой транслятором — Ассемблер.

Наиболее распространёнными ассемблерами для архитектуры x86 являются: • для DOS/Windows: Borland Turbo Assembler (TASM), Microsoft Macro Assembler (MASM) и Watcom assembler (WASM); • для GNU/Linux: gas (GNU Assembler), использующий AT&T-синтаксис, в отличие от большинства других популярных ассемблеров, которые используют Intel-синтаксис.

В процессе создания ассемблерной программы можно выделить четыре шага: набор текста, трансляция, компановка (линковка) и запуск программы. Для создания программ на языке ассемблера обычно пользуются утилитами командной строки.

4 Выполнение лабораторной работы

1) Для того, чтобы проделать лабораторную работу, мне необходимо создать каталог для работы с программами на языке ассемблера NASM. Перехожу в созданный каталог. При помощи утилиты touch создаю в каталоге текстовый файл с именем hello.asm и открываю при помощи gedit. Ввожу нужный текст.

```
srgubayjdullina@dk5n51 ~ $ mkdir -p ~/work/arch-pc/lab04
srgubayjdullina@dk5n51 ~ $ cd ~/work/arch-pc/lab04
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ touch hello.asm
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ gedit hello.asm
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $
```

Создание каталога. Создание тестового файла в каталоге и его компляция

2) Зная, что NASM превращает текст программы в объектный код, ввожу команду nasm, которая скомпилирует исходный файл hello.asm в obj.o. Тут же проверяю правильность введенной операции при помощи ls (рис.2)

```
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $
```

Компляция файла hello.asm

3) Для того, чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику ld. Сразу проверяю правильность выполненной операции (рис.3).

```
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Компановка файла hello.o

Выполняю следующую команду для компановки файла obj.o (рис.4)

```
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

Компановка файла obj.o

Также при помощи ld –help я изучила формат командной строки LD (рис.5)

```
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ ld --help
Использование ld [параметры] файл…
Параметры:
  -a КЛЮЧЕВОЕ СЛОВО
                              Управление общей библиотекой для совместимости с HP
  -A АРХИТЕКТУРА, --architecture АРХИТЕКТУРА
                              Задать архитектуру
  -b ЦЕЛЬ, --format ЦЕЛЬ
                              Задать цель для следующих входных файлов
  -c ФАЙЛ, --mri-script ФАЙЛ
                              Прочитать сценарий компоновщика в формате MRI
  -d, -dc, -dp                Принудительно делать общие символы определёнными
  --dependency-file ФАЙЛ  Write dependency file
  --force-group-allocation    Принудительно удалить членов группы из групп
  -e АДРЕС, --entry АДРЕС
                              Задать начальный адрес
  -E, --export-dynamic        Экспортировать все динамические символы
  --no-export-dynamic         Отменить действие --export-dynamic
```

Формат командной строки ld

4) Набрав в командной строке ./hello я запускаю на выполнение созданный исполняемый файл, находящийся в текущем каталоге (рис.6)

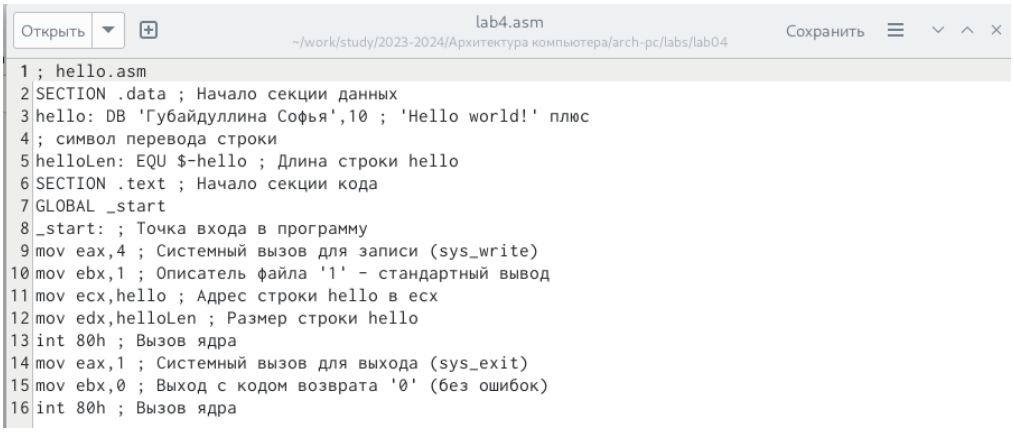
```
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ ./hello
Hello world!
```

Hello World!

5) Далее приступаю к выполнению самостоятельной работы. Для этого в новом каталоге ~/work/arch-pc/lab04 с помощью команды cp создаю копию файла hello.asm с именем lab4.asm (рис.7). При помощи gedit по заданию я меняю текст программы в lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с моими фамилией и именем. (рис.8)

```
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ gedit hello.asm
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ gedit lab4.asm
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $
```

Копирование файла hello.asm с именем lab4.asm



Редактирование в gedit

После мне необходимо транслировать полученный текст программы lab4.asm в объектный файл (рис.9) (рис.99).

```
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
```

Трансляция

```
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list.lst  main  obj.o
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $
```

Проверка выполненной трансляции

Компаную его и запускаю получившийся исполняемый файл при помощи изученных в ходе лабораторной работы операций (рис.10) (рис.11)

```
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab.4
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ ls
a.out  hello  hello.asm  hello.o  lab.4  lab4.asm  lab4.o  list.lst  main  obj.o
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ rm a.out
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab.4  lab4.asm  lab4.o  list.lst  main  obj.o
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $
```

Компановка

```
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $ ./lab.4
Губайдуллина Софья
srgubayjdullina@dk5n51 ~/work/arch-pc/lab04 $
```

Запуск исполняемого файла

В завершении копирую файлы hello.asm и lab4.asm в свой локальный репозиторий GitHub в привычный каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/ (рис.12), удаляя при этом остальные файлы, которые больше мне не пригодятся при помощи утилиты rm (рис.13)

```
srgubayjdullina@dk5n58 ~/work/arch-pc/lab04 $ cp * ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/lab
s/lab04/
srgubayjdullina@dk5n58 ~/work/arch-pc/lab04 $ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/
lab04/
srgubayjdullina@dk5n58 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello  hello.asm  hello.o  lab.4  lab4.asm  lab4.o  list.lst  main  obj.o  presentation  report
```

Копирование файлов в нужный каталог

```
srgubayjdullina@dk5n58 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ rm hello hello.o lab
.4 lab4.o list.lst main onj.o
rm: невозможно удалить 'onj.o': Нет такого файла или каталога
srgubayjdullina@dk5n58 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ rm obj.o
srgubayjdullina@dk5n58 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello.asm  lab4.asm  presentation  report
srgubayjdullina@dk5n58 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $
```

Удаление ненужных скопированных файлов

Полученные файлы загружаю в свой репозиторий GitHub (рис.14)

```
srgubayjdullina@dk5n58 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ git commit -am 'Add files for lab04'
[master 1891d03] Add files for lab04
 5 files changed, 33 insertions(+), 125 deletions(-)
 delete mode 100644 labs/lab03/report/image/placeimg_800_600_tech.jpg
 delete mode 100644 labs/lab03/report/report.docx
 delete mode 100644 labs/lab03/report/report.md
 create mode 100644 labs/lab04/hello.asm
 create mode 100644 labs/lab04/lab4.asm
srgubayjdullina@dk5n58 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ git push
Перечисление объектов: 15, готово.
Подсчет объектов: 100% (15/15), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (9/9), готово.
Запись объектов: 100% (9/9), 1.21 КиБ | 1.21 МиБ/с, готово.
Всего 9 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To github.com:srgubayjdullina/study_2023-2024_arh-pc.git
   213a194..1891d03  master -> master
```

Загрузка файлов в GitHub

5 Выводы

В ходе выполнения лабораторной работы я успешно освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.