Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютера

Мурзаев Замир Зейнадинович НБИбд-02-22

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Целью работы является освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Задание

1. В каталоге ~/work/arch-pc/lab05 с помощью команды cp создайте копию файла hello.asm с именем lab5.asm
2. С помощью любого текстового редактора внесите изменения в текст программы в файле lab5.asm так, чтобы вместо Hello world! на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы lab5.asm в объектный файл. Выполните компоновку объектного файла и запустите получивший- ся исполняемый файл.
4. Скопируйте файлы hello.asm и lab5.asm в Ваш локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/arch-pc/labs/lab05/. Загрузите файлы на Github.

3 Теоретическое введение

Процесс создания ассемблерной программы можно изобразить в виде следующей схемы **??**

В процессе создания ассемблерной программы можно выделить четыре шага:

• Набор текста программы в текстовом редакторе и сохранение её в от- дельном файле. Каждый файл имеет свой тип (или расширение), который определяет

назначение файла. Файлы с исходным текстом программ на языке ассемблера имеют тип asm.

• Трансляция — преобразование с помощью транслятора, например nasm, текста программы в машинный код, называемый объектным. На данном этапе также может быть получен листинг программы, содержащий кроме текста программы различную дополнительную информацию, созданную транслятором. Тип объектного файла — o, файла листинга — lst.

• Компоновка или линковка — этап обработки объектного кода компоновщиком (ld), который принимает на вход объектные файлы и собирает по ним исполняемый файл. Исполняемый файл обычно не имеет расшире- ния. Кроме того, можно получить файл карты загрузки программы в ОЗУ, имеющий расширение map.

• Запуск программы. Конечной целью является работоспособный испол- няемый файл. Ошибки на предыдущих этапах могут привести к некор- ректной работе программы, поэтому может присутствовать этап отладки программы при помощи специальной программы — отладчика. При на- хождении ошибки необходимо провести коррекцию программы, начиная с первого шага.

4 Выполнение лабораторной работы

Создаем новый каталог, как на рисунке 4.1, но в нашем случае каталог уже существует, поэтому терминал и выдаёт ошибку.

Рис. 4.1: Команда mkdir

Переходим в данный каталог, создаём в нём текстовый файл и открываем его через текстовый редактор gedit, как на рис 4.2.

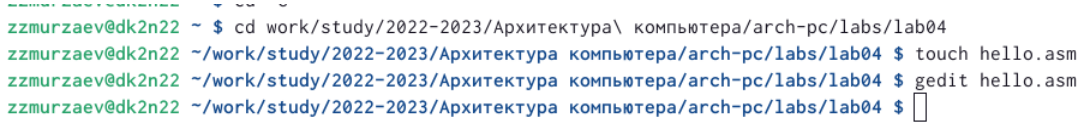

Рис. 4.2: создание текстового файла

Вводим в файл следующий текст: 4.3

```
; hello.asm
SECTION .data
hello: DB 'Hello, world',10

helloLen: EQU $-hello

SECTION .text
GLOBAL _start

_start:
mov eax,4
mov ebx,1
mov ecs,hello
mov edx,helloLen
int 80h

mov eax,1
mov ebx,0
int 80h
```

Рис. 4.3: написание текстовой программы на языке NASM

Скомпилируем текст написанной программы с помощью команды как на рис 4.4

Рис. 4.4: компилияция текста и создание объектного файла

Проверка успешности совершения команды на рис 4.5

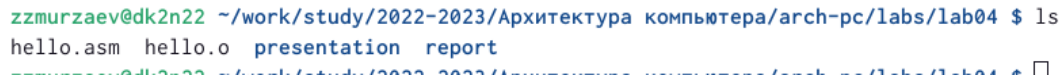

Рис. 4.5: команда ls

Скомпилируем тот же самый файл, только теперь зададим сами имя для объектного файла murzaev.o, как на рис 4.6

Рис. 4.6: опция -g - включение симолов для откладки, а опция -l - создание файла листинга

Компановка программы на рис 4.7 и последующая порверка с помощью ls

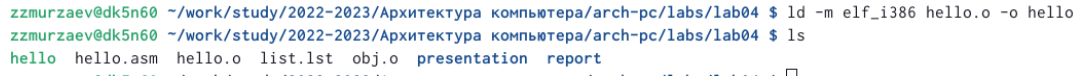

Рис. 4.7: команда ld

Сделаем то же самое с другими файлами как на рис 4.8. Исполняемый файл будет иметь имя main, а объектный - obj.o

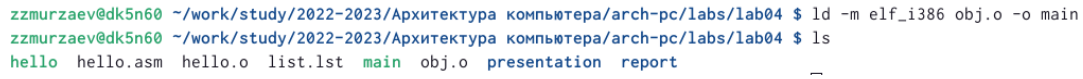

Рис. 4.8: команда ld

Формат командно строки LD можно увидеть, если ввести ld –help, как на рисунке 4.9

```
hello  hello.asm  hello.o  list.lst  main  obj.o  presentation  report
zzmurzaev@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld --help
Использование ld [параметры] файл…
Параметры:
  -a КЛЮЧЕВОЕ СЛОВО
                              Управление общей библиотекой для совместимости с HP/UX
  -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
  --enable-non-contiguous-regions
                              Enable support of non-contiguous memory regions
  --enable-non-contiguous-regions-warnings
                              Enable warnings when --enable-non-contiguous-regions may cause unexpected
behaviour
```

Рис. 4.9: команда ld –help

Чтобы запустить исполняемый файл, нужно набрать в командной строке ./и имя файла, смотреть на рис 4.10

```
zzmurzaev@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ./hello
Hello, world
```

Рис. 4.10: команда запуска файла

5 Задания самостоятельной работы

1)С помощью команды cp создайте копию файла hello.asm с именем lab4.asm, смотреть на рис 5.1

```
zzmurzaev@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ cp hello.asm lab4.asm
zzmurzaev@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o  presentation  report
```

Рис. 5.1: копирование файла

2)С помощью текстового редактора gedit 5.2 изменяю текст в файле так, чтобы вместо Hello world! выводились мои имя и фамилия 5.3

```
zzmurzaev@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ gedit lab4.asm
zzmurzaev@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 5.2: вызов текстового редактора

```
1  ; hello.asm
2  SECTION .data
3  hello: DB 'Murzaev Zamir',10
4
5  helloLen: EQU $-hello
6
7  SECTION .text
8  GLOBAL _start
9
10 _start:
11 mov eax,4
```

Рис. 5.3: замена текста в файле

3)Компилируем объектный файл, выполняем компоновку объектного файла и запускаем получившийся исполняемый файл, 5.4

```
zzmurzaev@dk3n33 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -o murzaev.o -f elf -g -l yapirak.lst lab4.asm
zzmurzaev@dk3n33 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 murzaev.o -o murzaev
zzmurzaev@dk3n33 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ./murzaev
Murzaev Zamir
```

Рис. 5.4: Процесс преобразования текстового файла в файл исполняемый и его запуск

Нам осталось только скопировать получившиеся файлы в локальный репозиторий и запустить на github. Весь процесс отображен на рис 5.5

```
zzmurzaev@dk3n33 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ git add .
zzmurzaev@dk3n33 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ git commit -am '(feat main): make course structure'
[master caf8104] (feat main): make course structure
 10 files changed, 78 insertions(+)
 create mode 100755 labs/lab04/hello
 create mode 100644 labs/lab04/hello.asm
 create mode 100644 labs/lab04/hello.o
 create mode 100644 labs/lab04/lab4.asm
 create mode 100644 labs/lab04/list.lst
 create mode 100755 labs/lab04/main
 create mode 100755 labs/lab04/murzaev
 create mode 100644 labs/lab04/murzaev.o
 create mode 100644 labs/lab04/obj.o
 create mode 100644 labs/lab04/yapirak.lst
zzmurzaev@dk3n33 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ git push
Перечисление объектов: 17, готово.
Подсчет объектов: 100% (17/17), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (14/14), готово.
Запись объектов: 100% (14/14), 3.12 КиБ | 1.56 МиБ/с, готово.
Всего 14 (изменений 7), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (7/7), completed with 2 local objects.
To github.com:Mintatar/study_2022-2023_arh-pc.git
   26245f7..caf8104  master -> master
```

Рис. 5.5: загрузка всех изменений на github

6 Выводы

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

Список литературы