

Лабораторная работа №4. Создание и процесс обработки программ на языке ассемблера NASM

Дисциплина: Архитектура компьютера

Козлова Нонна Юрьевна

Содержание

| | | |
|---|--------------------------------|----|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 4 | Выводы | 13 |

Список иллюстраций

| | | |
|-----|--|----|
| 3.1 | С помощью ls проверяем наличие объектного файла | 7 |
| 3.2 | С помощью ls проверяем наличие сохданных файлов | 8 |
| 3.3 | Используем команду nasm -hf | 9 |
| 3.4 | Используем команду man nasm | 10 |
| 3.5 | Чтобы посмотреть формат командной строки LD, нужно набрать ld -help | 11 |
| 3.6 | Набираем ./hello | 11 |
| 3.7 | Выполняем аналогично прошлым действиям | 12 |

Список таблиц

1 Цель работы

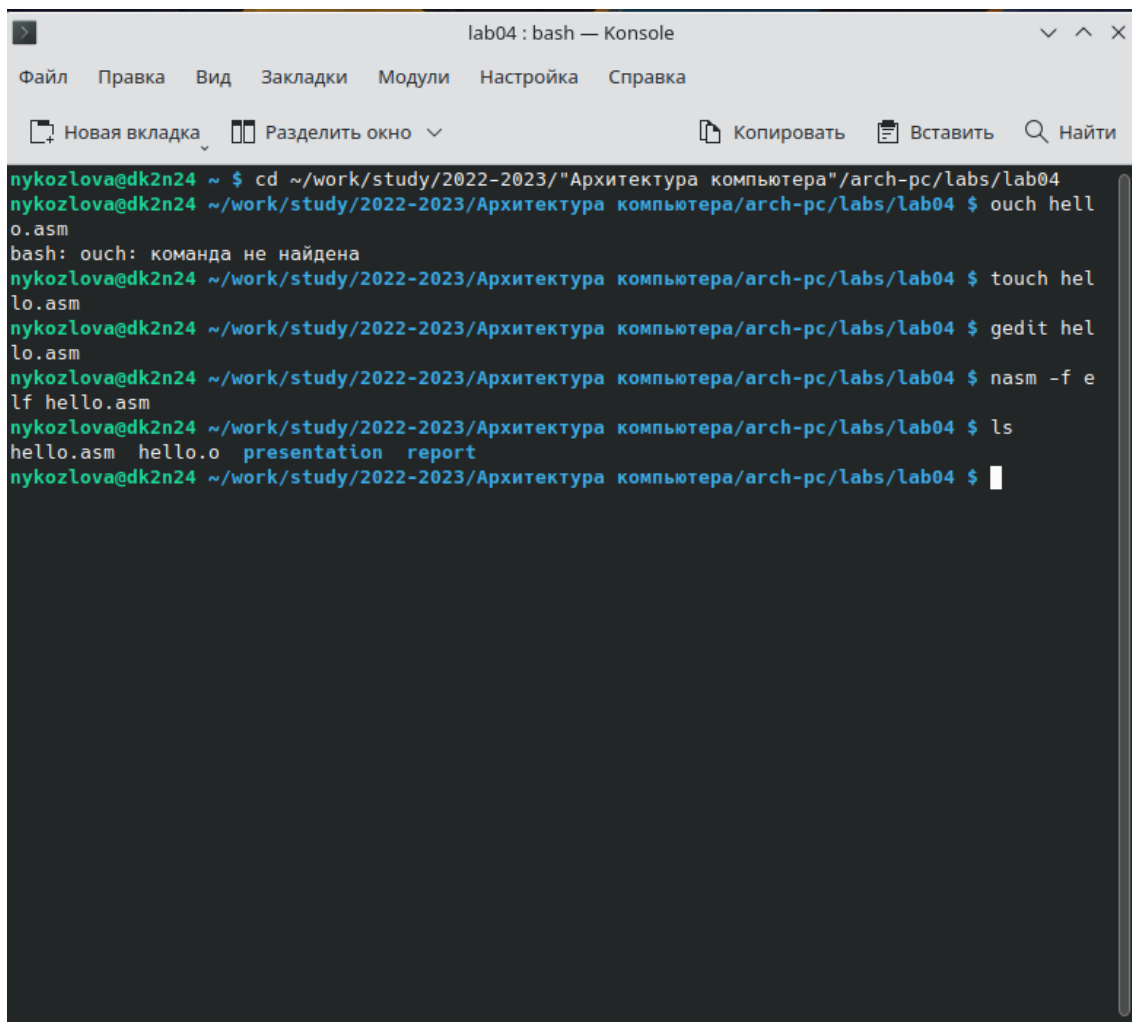
Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Задание

Вместо Hello world! на экран вывести строку с фамилией и именем. Создать отчет по лабораторной работе №4 в Markdown.

3 Выполнение лабораторной работы

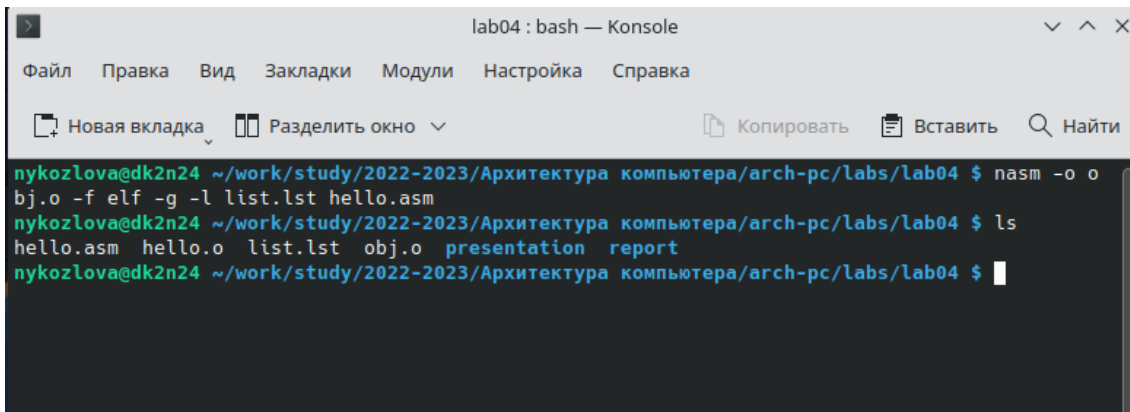
1. Создаем текстовый файл с именем hello.asm, открываем его с помощью текстового редактора gedit и вводим текст программы. Далее компилируем этот текст. (рис. 3.1)



```
lab04 : bash — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Найти
nykozlova@dk2n24 ~ $ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab04
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ touch hello.asm
bash: touch: команда не найдена
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ touch hello.asm
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ gedit hello.asm
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -f elf hello.asm
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello.asm hello.o presentation report
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.1: С помощью ls проверяем наличие объектного файла

2. Скомпилируем исходный файл hello.asm в obj.o (рис. 3.2)



```
lab04 : bash — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Найти
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -o o
bj.o -f elf -g -l list.lst hello.asm
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o  presentation  report
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.2: С помощью ls проверяем наличие созданных файлов

3. Получим список форматов объектного файла. (рис. 3.3)


```

nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -hf
Usage: nasm [-@ response_file] [options...] [--] filename
        nasm -v (or --v)

Options (values in brackets indicate defaults):

-h          show this text and exit (also --help)
-v (or --v) print the NASM version number and exit
-@ file     response file; one command line option per line

-o outfile  write output to outfile
--keep-all  output files will not be removed even if an error happens

-Xformat    specify error reporting format (gnu or vc)
-s          redirect error messages to stdout
-Zfile      redirect error messages to file

-M          generate Makefile dependencies on stdout
-MG         d:o, missing files assumed generated
-MF file    set Makefile dependency file
-MD file    assemble and generate dependencies
-MT file    dependency target name
-MQ file    dependency target name (quoted)
-MP         emit phony targets

-f format   select output file format
  bin       Flat raw binary (MS-DOS, embedded, ...) [default]
  ith       Intel Hex encoded flat binary
  srec      Motorola S-records encoded flat binary
  aout      Linux a.out
  aoutb     NetBSD/FreeBSD a.out
  coff      COFF (i386) (DJGPP, some Unix variants)
  elf32     ELF32 (i386) (Linux, most Unix variants)
  elf64     ELF64 (x86-64) (Linux, most Unix variants)
  elfx32    ELFX32 (ELF32 for x86-64) (Linux)
  as86      as86 (bin86/dev86 toolchain)
  obj       Intel/Microsoft OMF (MS-DOS, OS/2, Win16)
  win32     Microsoft extended COFF for Win32 (i386)
  win64     Microsoft extended COFF for Win64 (x86-64)
  ieee      IEEE-695 (LADsoft variant) object file format
  macho32   Mach-O i386 (Mach, including MacOS X and variants)
  macho64   Mach-O x86-64 (Mach, including MacOS X and variants)
  dbg       Trace of all info passed to output stage
  elf       Legacy alias for "elf32"
  macho     Legacy alias for "macho32"
  win       Legacy alias for "win32"

-g          generate debugging information
-F format   select a debugging format (output format dependent)
-gformat    same as -g -F format
  elf32:    dwarf   ELF32 (i386) dwarf (newer) [default]
           stabs   ELF32 (i386) stabs (older)

```

Рис. 3.3: Используем команду `nasm -hf`

4. Получим более подробную информацию (рис. 3.4)

```
NASM(1)                                The Netwide Assembler Project                                NASM(1)

NAME
    nasm - the Netwide Assembler, a portable 80x86 assembler

SYNOPSIS
    nasm [-@ response file] [-f format] [-o outfile] [-l listfile] [options...]
    filename

DESCRIPTION
    The nasm command assembles the file filename and directs output to the file outfile
    if specified. If outfile is not specified, nasm will derive a default output file
    name from the name of its input file, usually by appending '.o' or '.obj', or by
    removing all extensions for a raw binary file. Failing that, the output file name
    will be 'nasm.out'.

OPTIONS
    -@ filename
        Causes nasm to process options from filename as if they were included on the
        command line.

    -a
        Causes nasm to assemble the given input file without first applying the macro
        preprocessor.

    -D|-d macro[=value]
        Pre-defines a single-line macro.

    -E|-e
        Causes nasm to preprocess the given input file, and write the output to stdout
        (or the specified output file name), and not actually assemble anything.

    -f format
        Specifies the output file format. To see a list of valid output formats, use
        the -hf option.

    -F format
        Specifies the debug information format. To see a list of valid output formats,
        use the -y option (for example -felf -y).

    -g
        Causes nasm to generate debug information.

    -gformat
        Equivalent to -g -F format.

    -h
        Causes nasm to exit immediately, after giving a summary of its invocation
        options.

    -hf
        Manual page nasm(1) line 1/290 14% (press h for help or q to quit)
```

Рис. 3.4: Используем команду `man nasm`

5. Передаем объектный файл на обработку компоновщику и посмотрим на формат командной строки LD (рис. 3.5)

```

nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 hello.o -o hello
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello hello.asm hello.o list.lst obj.o presentation report
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 obj.o -o main
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld --helpИспользование 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
-EB                               Компоновать объекты с прямым порядком байтов
-EL                               Компоновать объекты с обратным порядком байтов
-f SHLIB, --auxiliary SHLIB       Вспомогательный фильтр таблицы символов общих объектов
-F SHLIB, --filter SHLIB          Фильтр для таблицы символов общих объектов
-g                               Игнорируется
-G РАЗМЕР, --gpsize РАЗМЕР        Размер маленьких данных (если не указан, то берётся из --shared)
-h ИМЯ_ФАЙЛА, -soname ИМЯ_ФАЙЛА  Задать внутреннее имя общей библиотеки
-I ПРОГРАММА, --dynamic-linker ПРОГРАММА      Назначить ПРОГРАММУ в качестве используемого динамического компоновщика
--no-dynamic-linker              Создать исполняемый файл без заголовка программного интерпретатора
-l LIBNAME, --library LIBNAME     Искать библиотеку с именем LIBNAME
-L КАТАЛОГ, --library-path КАТАЛОГ        Добавить КАТАЛОГ к пути поиска библиотек
--sysroot=<DIRECTORY>           Заменить расположение по умолчанию sysroot
-m ЭМУЛЯЦИЯ                      Задать эмуляцию
-M, --print-map                  Напечатать файл карты на стандартном выводе
-n, --nmagic                     Не выравнивать данные по границе страницы
-N, --omagic                     Не выравнивать данные по границе страницы, не делать сегмент текста доступным только для чтения
--no-omagic                     Выравнивать данные по границе страницы, сделать сегмент текста доступным только для чтения
-o ФАЙЛ, --output ФАЙЛ          Задать имя выходного файла
-O                               Оптимизировать выходной файл
--out-implib ФАЙЛ               Генерировать импортируемую библиотеку
-plugin МОДУЛЬ                  Загрузить модуль с указанным именем

```

Рис. 3.5: Чтобы посмотреть формат командной строки LD, нужно набрать `ld -help`

6. Запускаем на выполнение созданный исполняемый файл. (рис. 3.6)

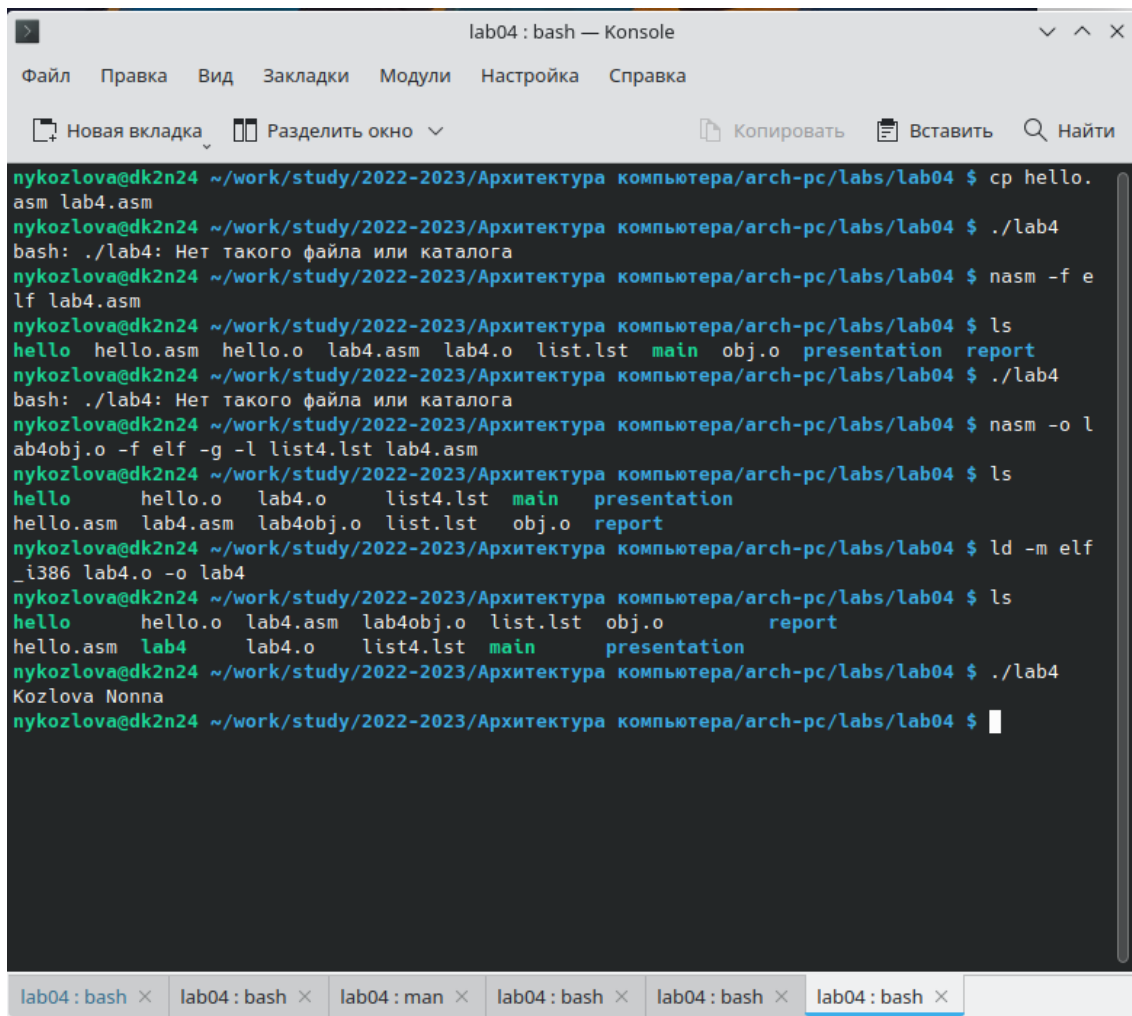
```

nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ./hello
Hello world!
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ █

```

Рис. 3.6: Набираем `./hello`

7. Делаем так, чтобы на экран выводилась строка с именем и фамилией. (рис. 3.7)



```
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ cp hello.asm lab4.asm
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ./lab4
bash: ./lab4: Нет такого файла или каталога
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -f elf lab4.asm
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list.lst  main  obj.o  presentation  report
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ./lab4
bash: ./lab4: Нет такого файла или каталога
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -o lab4obj.o -f elf -g -l list4.lst lab4.asm
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello  hello.o  lab4.o  list4.lst  main  presentation
hello.asm  lab4.asm  lab4obj.o  list.lst  obj.o  report
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 lab4.o -o lab4
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello  hello.o  lab4  lab4.asm  lab4obj.o  list.lst  obj.o  report
hello.asm  lab4  lab4.o  list4.lst  main  presentation
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ./lab4
Kozlova Nonna
nykozlova@dk2n24 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.7: Выполняем аналогично прошлым действиям

4 Выводы

В ходе лабораторной работы я поняла принцип сборки программ, написанных на ассемблере NASM. # Список литературы{.unnumbered}