

Лабораторная работа №4

Создание и процесс обработки программ на языке ассемблера NASM

Дмитрий Сергеевич Хохлов

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	10
	Список литературы	11

Список иллюстраций

2.1	Создание каталога lab04	6
2.2	Переход в каталог	6
2.3	Создание файла hello.asm	6
2.4	Открытие файла hello.asm	6
2.5	Ввод текста	7
2.6	Компиляция программы “Hello world”	7
2.7	Компиляция файла hello.asm в obj.o	7
2.8	Передача объектного файла компоновщику LD	7
2.9	Ввод команды	7
2.10	Запуск файла	8
2.11	Создание копии файла helo.asm с именем lab04.asm	8
2.12	Вношу изменения в текст программы в файле lab04.asm	8
2.13	Вношу изменения в текст программы в файле lab04.asm	9
2.14	Запуск файла	9
2.15	Копирование файлов и загрузка их на GitHub	9

Список таблиц

1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

2 Выполнение лабораторной работы

Создаю каталог для работы с программами на языке ассемблера NASM (рис. [2.1]).

```
dskhokhlov@dk4n65 ~ $ mkdir -p ~/work/arch-pc/lab04
```

Рис. 2.1: Создание каталога lab04

Перехожу в созданный каталог (рис. [2.2]).

```
dskhokhlov@dk4n65 ~ $ cd ~/work/arch-pc/lab04
```

Рис. 2.2: Переход в каталог

Создаю файл с именем hello.asm (рис. [2.3]).

```
dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ touch hello.asm
```

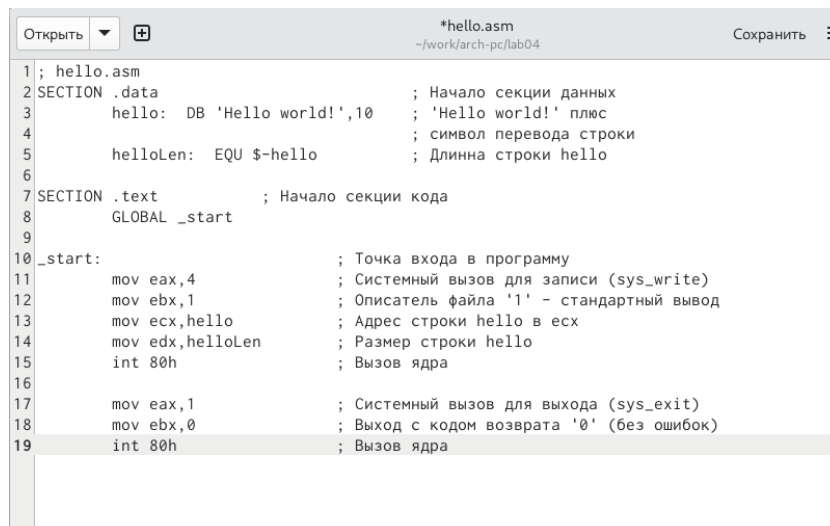
Рис. 2.3: Создание файла hello.asm

Открываю файл с помощью текстового редактора gedit (рис. [2.4]).

```
dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Рис. 2.4: Открытие файла hello.asm

Ввожу данные мне текст (рис. [2.5]).



```
1; hello.asm
2SECTION .data                ; Начало секции данных
3    hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4                                ; символ перевода строки
5    helloLen: EQU $-hello      ; Длина строки hello
6
7SECTION .text                ; Начало секции кода
8    GLOBAL _start
9
10 _start:
11        mov eax,4             ; Системный вызов для записи (sys_write)
12        mov ebx,1             ; Описатель файла '1' - стандартный вывод
13        mov ecx,hello         ; Адрес строки hello в ecx
14        mov edx,helloLen      ; Размер строки hello
15        int 80h               ; Вызов ядра
16
17        mov eax,1             ; Системный вызов для выхода (sys_exit)
18        mov ebx,0             ; Выход с кодом возврата '0' (без ошибок)
19        int 80h               ; Вызов ядра
```

Рис. 2.5: Ввод текста

Ввожу команду для компиляции текста программы “Hello world” (рис.6 [2.6]).

```
dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
```

Рис. 2.6: Компиляция программы “Hello world”

Компилирую исходный файл hello.asm в obj.o (рис. [2.7]).

```
dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Рис. 2.7: Компиляция файла hello.asm в obj.o

Передаю объектный файл компоновщику LD (рис. [2.8]).

```
dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
```

Рис. 2.8: Передача объектного файла компоновщику LD

Аналогично ввожу следующую команду (рис. [2.9]).

```
dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
```

Рис. 2.9: Ввод команды

Запускаю исполняемый файл (рис. [2.10]).

```
dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ ./hello
Hello world!
```

Рис. 2.10: Запуск файла

#Выполнение самостоятельной работы

В каталоге ~/work/arch-pc/lab04 с помощью команды `cp` создаю копию файла `hello.asm` с именем `lab04.asm` (рис. [2.11]).

```
dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ cp hello.asm lab04.asm
```

Рис. 2.11: Создание копии файла `hello.asm` с именем `lab04.asm`

С помощью текстового редактора `gedit` вношу изменения в текст программы `lab04.asm` так, чтобы вместе с `Hello world!` на экран выводилась строка с моими фамилией и именем (рис. [2.12]).

```
dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ gedit lab04.asm
dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ nasm -f elf lab04.asm
dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst lab04.asm
dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab04.o -o hello
dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
```

Рис. 2.12: Вношу изменения в текст программы в файле `lab04.asm`

Транслирую полученный текст программы `lab04.asm` в объектный файл. Выполняю компоновку объектного файла (рис. [2.13]).


```

1; hello.asm
2SECTION .data                ; Начало секции данных
3    hello: DB 'Hello world! Khokhlov Dmitry',10 ; 'Hello world!' плюс
4                ; символ перевода строки
5    helloLen: EQU $-hello    ; Длина строки hello
6
7SECTION .text                ; Начало секции кода
8    GLOBAL _start
9
10 _start:
11        mov eax,4            ; Точка входа в программу
12        mov ebx,1            ; Системный вызов для записи (sys_write)
13        mov ecx,hello        ; Описатель файла '1' - стандартный вывод
14        mov edx,helloLen     ; Адрес строки hello в ecx
15        int 80h              ; Размер строки hello
16                                ; Вызов ядра
17
18        mov eax,1            ; Системный вызов для выхода (sys_exit)
19        mov ebx,0            ; Выход с кодом возврата '0' (без ошибок)
20        int 80h              ; Вызов ядра

```

Рис. 2.13: Вношу изменения в текст программы в файле lab04.asm

Запускаю получившийся исполняемый файл (рис. [2.14]).

```

dskhokhlov@dk4n65 ~/work/arch-pc/lab04 $ ./hello
Hello world! Khokhlov Dmitry

```

Рис. 2.14: Запуск файла

Копирую файлы hello.asm и lab04.asm в свой локальный репозиторий и загружаю файлы на Github (рис. [2.15]).

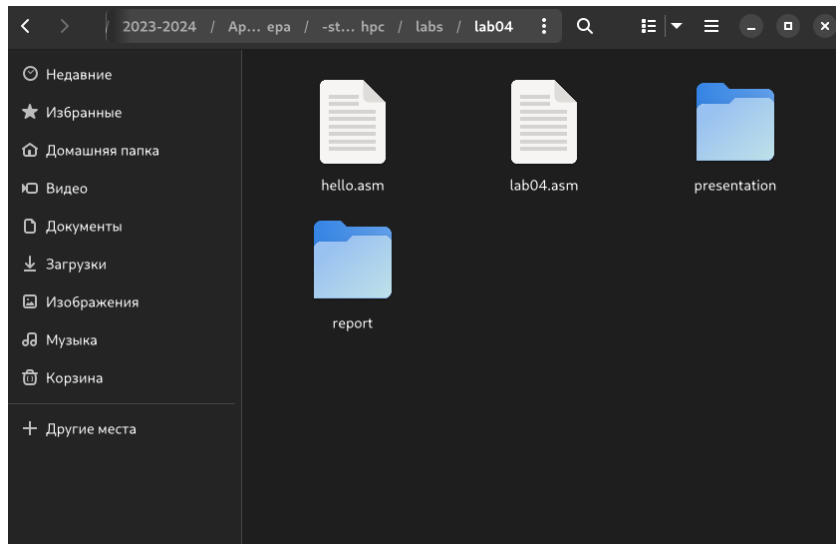


Рис. 2.15: Копирование файлов и загрузка их на GitHub

3 Выводы

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

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.

15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер,
17. — 1120 с. — (Классика Computer Science).