

Отчёта по лабораторной работе 4

Архитектура компьютера

Волчкова Елизавета Дмитриевна

Содержание

Цель работы

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

1. В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создайте копию файла `hello.asm` с именем `lab4.asm`
2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы `lab4.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы `hello.asm` и `lab4.asm` в Ваш локальный репозиторий в каталог `~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/`. Загрузите файлы на Github.

Теоретическое введение

Основной задачей процессора является обработка информации, а также организация координации всех узлов компьютера. В состав центрального процессора (ЦП) входят следующие устройства: • арифметико-логическое устройство (АЛУ) — выполняет логические и арифметические действия, необходимые для обработки информации, хранящейся в памяти; • устройство управления (УУ) — обеспечивает управление и контроль всех устройств компьютера; • регистры — сверхбыстрая оперативная память небольшого объёма, входящая в состав процессора, для временного хранения промежуточных результатов выполнения инструкций; регистры процессора делятся на два типа: регистры общего назначения и специальные регистры.

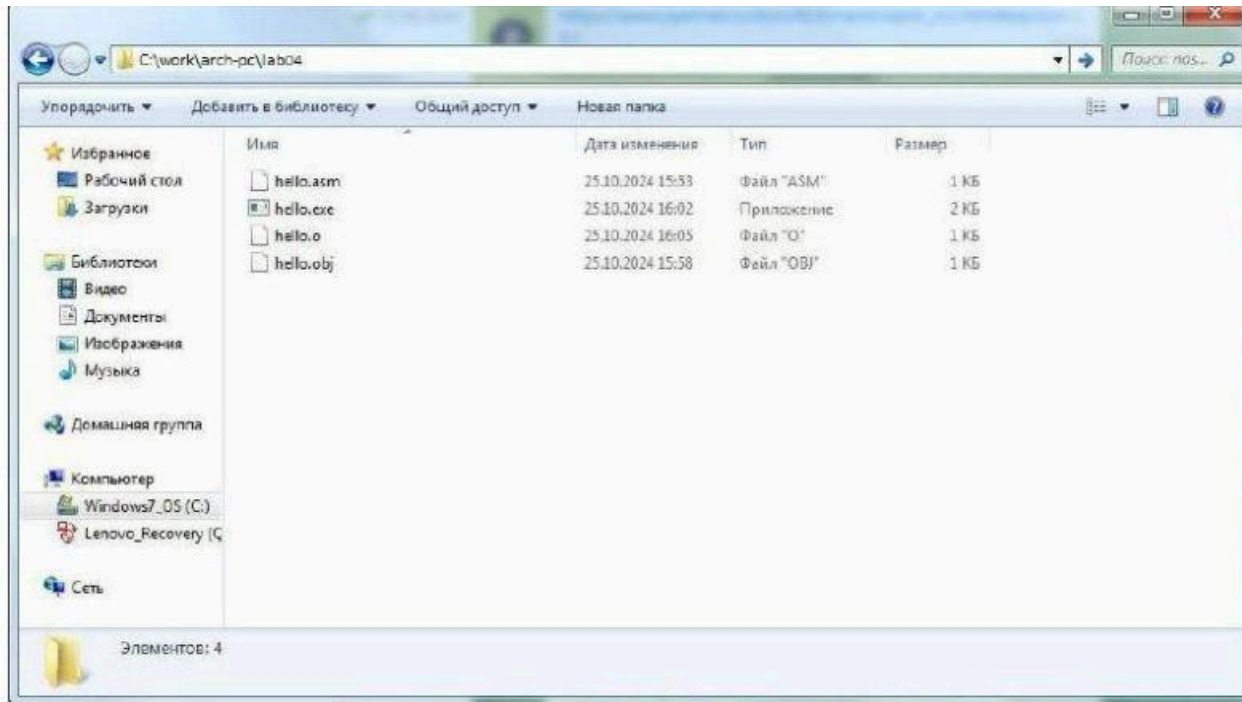
Выполнение лабораторной работы

Я воспользовалась командой “mov” – команда пересылки данных на языке ассемблера. Возвела “Hello world!” - на экран. Команда mov переносит данные.

```
root@LAPTOP-KE7VHG2Q:~# mkdir -p ~/work/arch-pc/lab04
root@LAPTOP-KE7VHG2Q:~/lab04#
root@LAPTOP-KE7VHG2Q:~/lab04# touch hello.asm
root@LAPTOP-KE7VHG2Q:~/lab04#
root@LAPTOP-KE7VHG2Q:~/lab04# gedit hello.asm
```

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

Создала каталог для работы с программами на языке ассемблера NASM: mkdir -p
~/work/arch-pc/lab04 Затем перешла в созданный каталог cd
~/work/arch-pc/lab04 Далее создала текстовый файл с именем hello.asm touch hello.asm



Открыла этот файл с помощью текстового редактора, например, gedit и ввела в него следующий текст: `; hello.asm`
`SECTION .data`; Начало секции данных
`hello: DB 'Hello world!',10 ; 'Helloworld!'` плюс; символ перевода строки
`helloLen: EQU $-hello` ; Длина строки
`SECTION .text` ; Начало секции кода
`GLOBAL _start` ;
Затем ввела точку входа в программу `mov eax,4` ; Системный вызов для записи (sys_write) `mov ebx,1` ; Потом описала файл '1' - стандартный вывод `mov ecx,hello` ; [Адрес строки hello в ecx `mov edx,helloLen` ; Размер строки hello `int 80h` ; Вызов ядра `mov eax,1` ;] После сделала следующие действия: Системный вызов для выхода (sys_exit) `mov ebx,0` ; Выход с кодом возврата '0' (без ошибок) `int 80h` ; Вызов ядра
Ввела полный вариант командной строки: Выполнила следующую команду: `nasm -o obj.o -f elf -g -l list.lst hello.asm`

С помощью команды ls проверила, что файлы были созданы. Далее выполнила следующую команду: `ld -melf_i386 obj.o -o main`

Выводы

Целью работы было -своение процедуры компиляции и сборки программ, написанных на ассемблере NASM, продеалав данные задания, я освоила процедуры в NASM. # Список литературы{unnumbered}

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/Learningbash-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., Lupin С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс,
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. — БХВПетербург, 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).