

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

дисциплина: Архитектура компьютера

Студент: Бусардыков Артур Валерьевич

Группа: НпМБВ-02-21

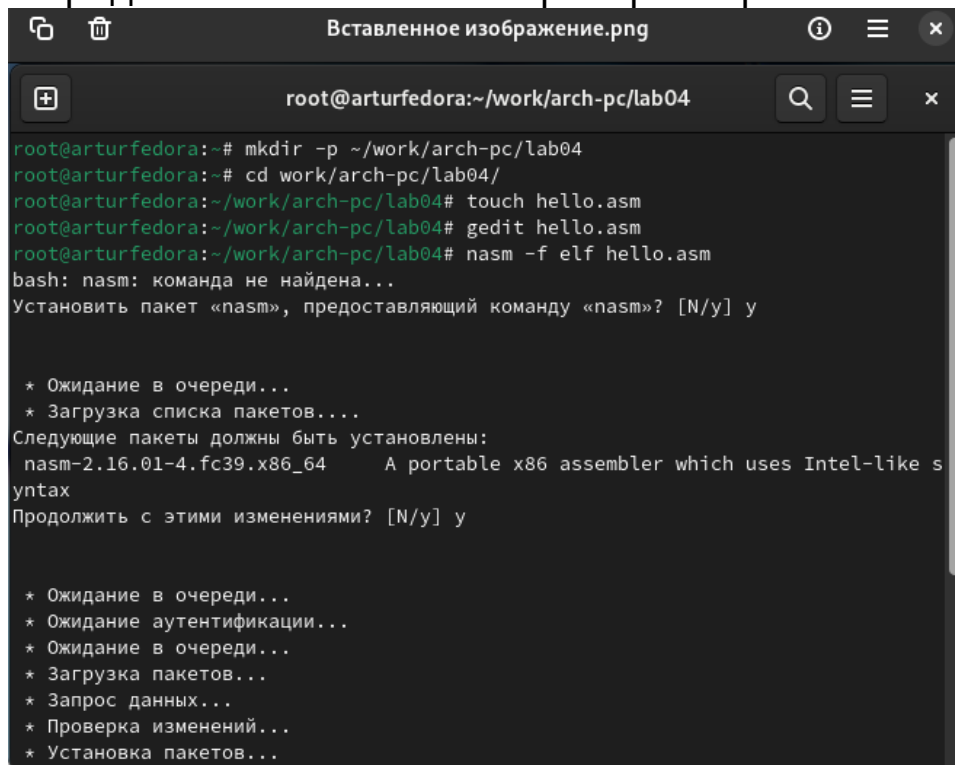
МОСКВА

2023 г.

Цель работы:

Изучить процедуры компиляции и сборки программ, написанных на языке ассемблере NASM

Порядок выполнения лабораторной работы:

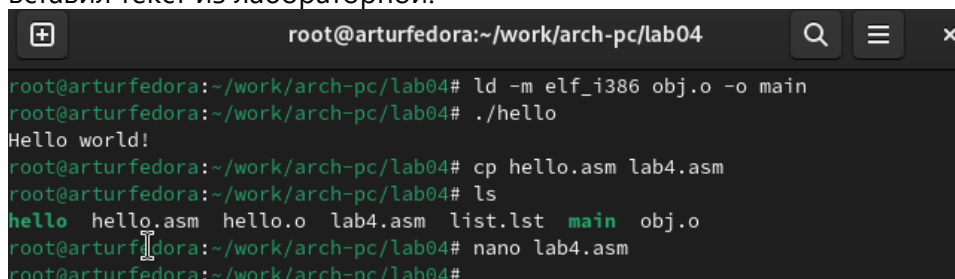


```
root@arturfedora:~/work/arch-pc/lab04
root@arturfedora:~# mkdir -p ~/work/arch-pc/lab04
root@arturfedora:~# cd work/arch-pc/lab04/
root@arturfedora:~/work/arch-pc/lab04# touch hello.asm
root@arturfedora:~/work/arch-pc/lab04# gedit hello.asm
root@arturfedora:~/work/arch-pc/lab04# nasm -f elf hello.asm
bash: nasm: команда не найдена...
Установить пакет «nasm», предоставляющий команду «nasm»? [N/y] y

* Ожидание в очереди...
* Загрузка списка пакетов....
Следующие пакеты должны быть установлены:
  nasm-2.16.01-4.fc39.x86_64      A portable x86 assembler which uses Intel-like s
yntax
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...
```

Создал каталог, перешел в него, создал файл hello.asm. Открыл его на редактирование и вставил текст из лабораторной.



```
root@arturfedora:~/work/arch-pc/lab04# ld -m elf_i386 obj.o -o main
root@arturfedora:~/work/arch-pc/lab04# ./hello
Hello world!
root@arturfedora:~/work/arch-pc/lab04# cp hello.asm lab4.asm
root@arturfedora:~/work/arch-pc/lab04# ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
root@arturfedora:~/work/arch-pc/lab04# nano lab4.asm
root@arturfedora:~/work/arch-pc/lab04#
```

Далее

```
root@arturfedora:~/work/arch-pc/lab04
GNU nano 7.2 lab4.asm
hello.asm
SECTION .data ; Начало секции данных
hello: DB 'Busardykov Artur !',10 ; 'Busardykov Artur ' плюс
; символ перевода строки
helloLen: EQU $-Busardykov Artur; Длина строки Busardykov Artur
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

[ Прочитано 16 строк ]
G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выровнять ^/_ К строке
```

Надо его транслировать. Написал команду для компиляции. Выполнил команду с расширенным синтаксисом и проверил наличие файла, затем выполнил команду с ключом -o. Далее запустил файл с помощью ./hello.

Задание для самостоятельной работы.

Скопировал эти файлы в папку с репозиторием, затем открыл файл lab4.asm для редактирования.

```
root@arturfedora:~/work/arch-pc/lab04# gedit lab4.asm
root@arturfedora:~/work/arch-pc/lab04# nasm -f elf lab4.asm
root@arturfedora:~/work/arch-pc/lab04# ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
root@arturfedora:~/work/arch-pc/lab04# ld -m elf_i386 lab4.o -o lab04
root@arturfedora:~/work/arch-pc/lab04# ls
hello hello.asm hello.o lab04 lab4.asm lab4.o list.lst main obj.o
root@arturfedora:~/work/arch-pc/lab04# ld -m elf_i386 obj.o -o main
root@arturfedora:~/work/arch-pc/lab04# ./lab04
busardykov
```

Повторил шаги трансляции и компоновки.

Затем выполнил команду ./lab4.asm и вывелась Фамилия на экран.

Далее скопировал эти файлы в папку с лабораторными и запустил на гитхаб.

```
root@arturfedora:~/work/arch-pc/lab04# cp hello.asm ~/work/study/2023-2024/Архитектура\
компьютера/arch-pc/labs/lab04/
root@arturfedora:~/work/arch-pc/lab04# cp lab4.asm ~/work/study/2023-2024/Архитектура\
компьютера/arch-pc/labs/lab04/
root@arturfedora:~/work/arch-pc/lab04#
```

Вопросы для самопроверки

- 1) Отличия ассемблерных программ от программ на языках высокого уровня: ассемблерные программы ближе к аппаратуре, используют мнемокоды для команд процессора.
- 2) Инструкция в ассемблере - это команда процессору, директива - указание ассемблеру для обработки, но не выполняется процессором.
- 3) Основные правила оформления программ на ассемблере включают мнемокоды, метки, операнды, комментарии.
- 4) Этапы получения исполняемого файла: трансляция и компоновка.
- 5) Назначение этапа трансляции - преобразование ассемблерного кода в машинный код.
- 6) Назначение этапа компоновки - объединение объектных файлов и библиотек в исполняемый файл.
- 7) При трансляции могут создаваться объектные файлы (.o), исполняемые файлы (.exe), библиотеки (.lib); по умолчанию - объектные файлы.
- 8) Форматы файлов для `nasm`: `.asm` (исходный код), `.obj` (объектный файл); для `ld`: `.o` (объектный файл), `.exe` (исполняемый файл).

