



РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

Создание и процесс обработки программ на языке ассемблера NASM

Студент: Эйвази Мани

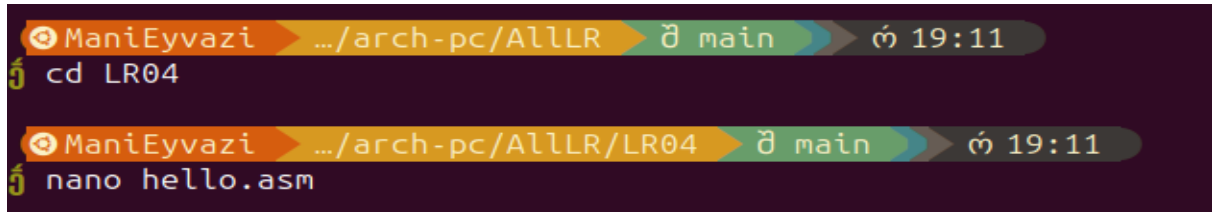
Группа: НПИбд-03-24

Студенческий билет №: 1032245107

Цель работы

Освоить процедуры компиляции и сборки программ, познакомиться с языком ассемблера NASM.

Первый шаг: Базовая настройка git: создаем текстовый файл на формате <<.asm>>



```
ManiEyvazi > .../arch-pc/AllLR > ð main > 19:11
$ cd LR04

ManiEyvazi > .../arch-pc/AllLR/LR04 > ð main > 19:11
$ nano hello.asm
```

PIC_01

Второй шаг: открываем файл и заполняем его по примеру

```
section .data
    hello db 'Hello, world!' , 10    ; The string to print , a new line

section .text
    global _start                    ; Entry point for the program

_start:
    ; Write the string to printout
    mov eax, 4                      ; syscall number for sys_write
    mov ebx, 1                      ; file descriptor 1 is stdout
    mov ecx, hello                  ; pointer to the string
    mov edx, 13                     ; size of the string
    int 0x80                        ; call the kernel

    ;exit program

    mov eax, 1                      ; syscall number for sys_exit
    mov ebx, 0                      ; return 0 status
    int 0x80                        ; call the kernel
```

PIC_02

Третий шаг: преобразуем текст программы в объектный код. (Используем команду `nasm`)

```
ManiEyvazi > .../arch-pc/AllLR/LR04 > ð main ? > 19:24
$ ls
hello.asm

ManiEyvazi > .../arch-pc/AllLR/LR04 > ð main ? > 19:24
$ nasm -f elf32 hello.asm -o hello.o

ManiEyvazi > .../arch-pc/AllLR/LR04 > ð main ? > 19:25
$ ls
hello.asm  hello.o
```

PIC03

Четвертый шаг: передаем объектный файл на обработку компоновщику

```
ManiEyvazi > .../arch-pc/AllLR/LR04 > ð main ? > 19:25
$ ld -m elf_i386 hello.o -o hello
```

PIC04

Пятый шаг: запускаем на выполнение созданный исполняемый файл

```
ManiEyvazi > .../arch-pc/AllLR/LR04 > ð main ? > 19:26
$ ./hello
Hello, world!
```

PIC05

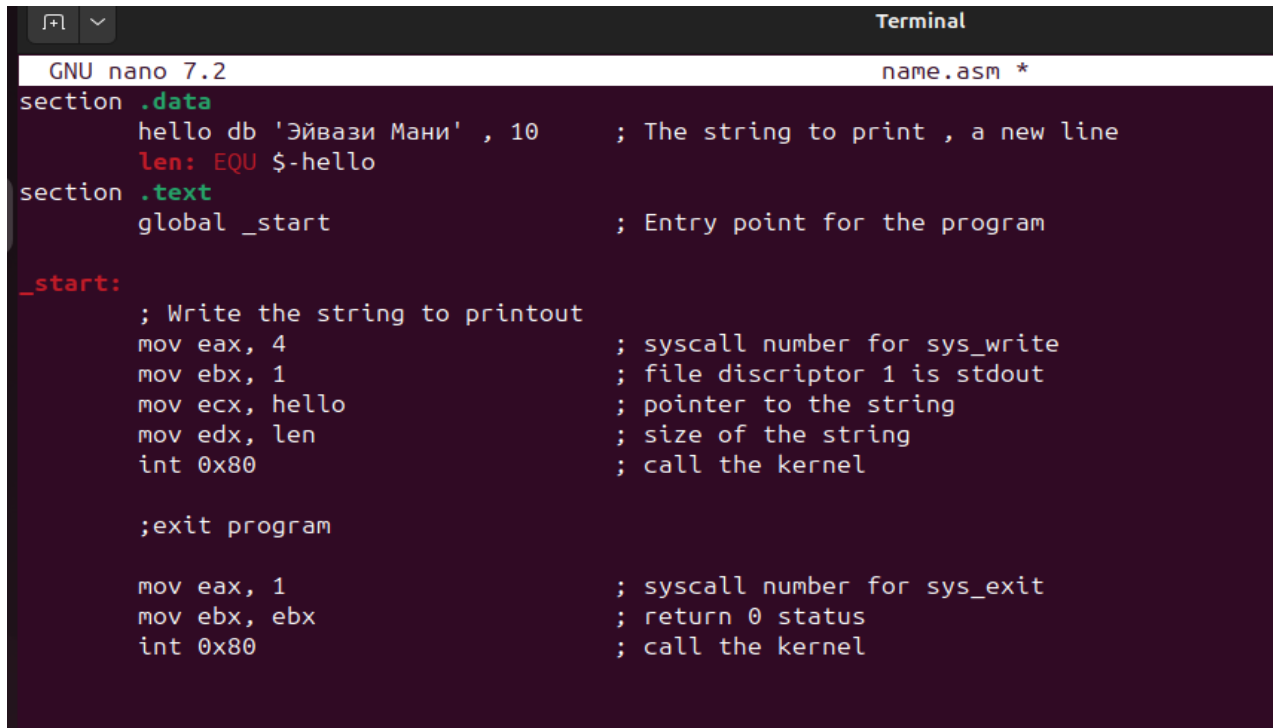
Шестой шаг: создаем копию файла `hello.asm`

```
ManiEyvazi > .../arch-pc/AllLR/LR04 > ð main > 19:40
$ cp hello.asm name.asm

ManiEyvazi > .../arch-pc/AllLR/LR04 > ð main ? > 19:40
$ nano name.asm
```

PIC06

Седьмой шаг: открываем файл и редактируем его



```
GNU nano 7.2 name.asm *
section .data
    hello db 'Эйвази Мани' , 10      ; The string to print , a new line
    len: EQU $-hello
section .text
    global _start                    ; Entry point for the program

_start:
    ; Write the string to printout
    mov eax, 4                      ; syscall number for sys_write
    mov ebx, 1                      ; file descriptor 1 is stdout
    mov ecx, hello                  ; pointer to the string
    mov edx, len                    ; size of the string
    int 0x80                        ; call the kernel

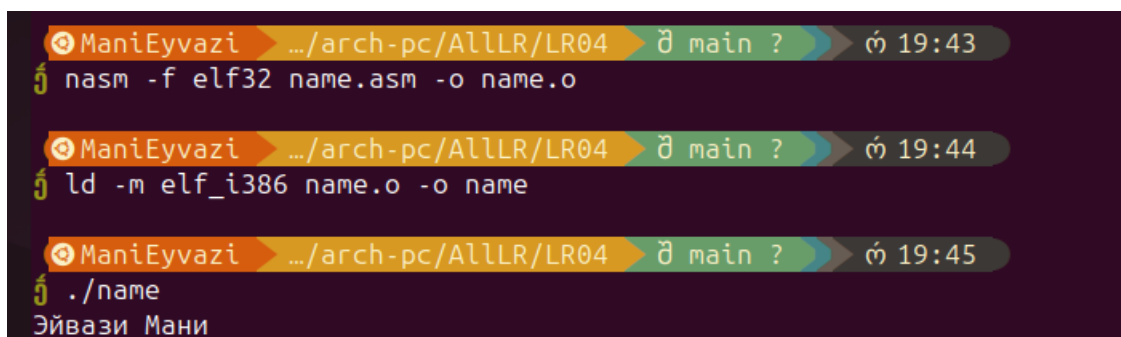
    ;exit program

    mov eax, 1                      ; syscall number for sys_exit
    mov ebx, ebx                    ; return 0 status
    int 0x80                        ; call the kernel
```

PIC07

(На этот раз вместо того, чтобы ставить в размер число 13, мы дали программу, чтобы система сама считала размер числа введённого текста <третья строка>)

Восьмой шаг: Прописываем те же команды, что и с первой программой



```
ManiEyvazi ~/arch-pc/AllLR/LR04 > ð main ? 19:43
$ nasm -f elf32 name.asm -o name.o

ManiEyvazi ~/arch-pc/AllLR/LR04 > ð main ? 19:44
$ ld -m elf_i386 name.o -o name

ManiEyvazi ~/arch-pc/AllLR/LR04 > ð main ? 19:45
$ ./name
Эйвази Мани
```

PIC08

Последний шаг: переходим в каталог лабораторных работ и загружаем файлы на Github.

```
ManiEyvazi > .../arch-pc/AllLR/LR04 > ð main + > 20:00
$ git add .

ManiEyvazi > .../arch-pc/AllLR/LR04 > ð main + > 20:00
$ git commit -m 'Laboratornaya rabota no.4 all assembly files added'
[main b4626e5] Laboratornaya rabota no.4 all assembly files added
3 files changed, 19 insertions(+)
create mode 100755 AllLR/LR04/name
create mode 100644 AllLR/LR04/name.asm
create mode 100644 AllLR/LR04/name.o

ManiEyvazi > .../arch-pc/AllLR/LR04 > ð main ↑ > 20:00
$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.47 KiB | 1.47 MiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:ManiEyvazi/study_NPIbd03_arh-pc.git
d41d5e5..b4626e5  main -> main

ManiEyvazi > .../arch-pc/AllLR/LR04 > ð main > 20:00
$
```

PIC08

Заключение

Мы познакомились с языком ассемблера NASM и создали две работающих программы.