

Лабораторная работа №4

**Создание и процесс обработки программ на языке ассемблера
NASM**

Борисенкова София Павловна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение задания для самостоятельной работы	10
4	Выводы	12

Список иллюстраций

2.1	Создание .asm файла	6
2.2	Открытие созданного файла	6
2.3	Редактирование файла	7
2.4	Проверка успешного создания файла краткой командой	7
2.5	Проверка успешного создания файла полной командой	8
2.6	Проверка на успешное создание исполняемого файла	8
2.7	Проверка на успешное создание файла	8
2.8	Запуск исполняемого файла hello	9
3.1	Процесс редактирования файла	10
3.2	Запуск собранного файла	11

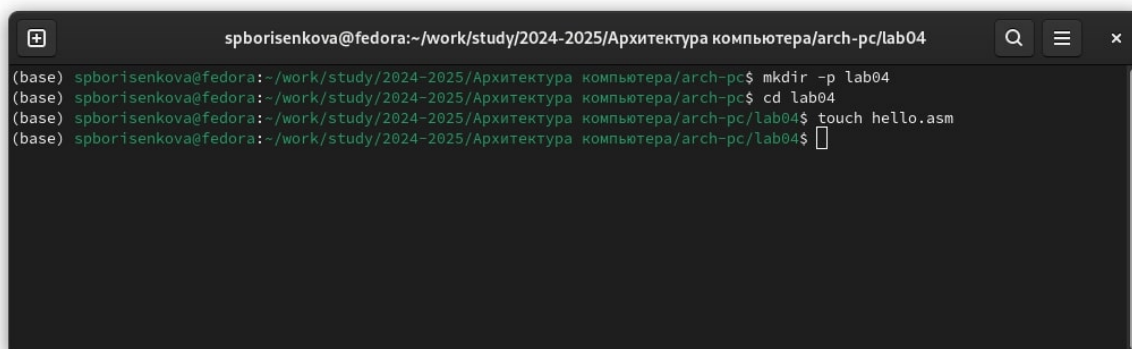
Список таблиц

1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

2 Выполнение лабораторной работы

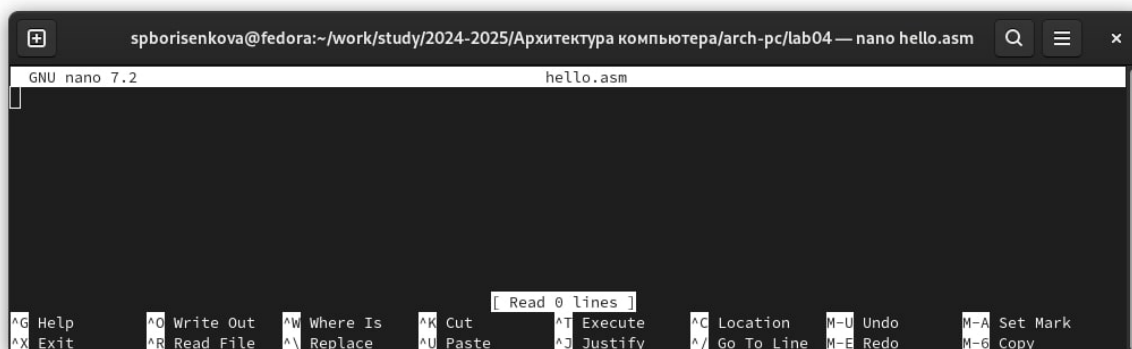
Создадим нужную директорию с помощью команды `mkdir`. Переместимся в созданный нами каталог и создадим файл `hello` с расширением `.asm`, в котором мы будем писать код на ассемблере (Рис. 2.1):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ mkdir -p lab04
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cd lab04
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$ touch hello.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$
```

Рис. 2.1: Создание `.asm` файла

Для того, чтобы редактировать созданный файл, воспользуемся текстовым редактором `nano` (Рис. 2.2):

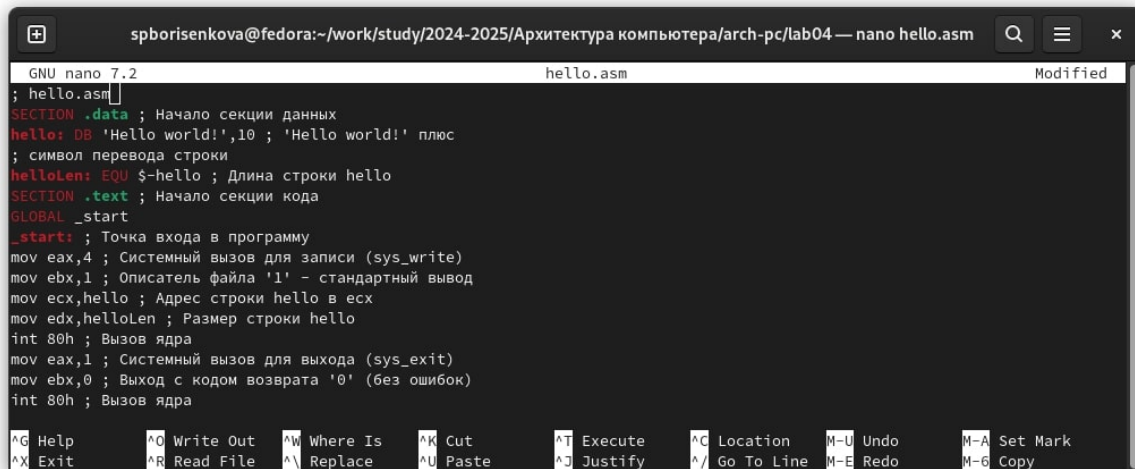


```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04 — nano hello.asm
GNU nano 7.2 hello.asm

[ Read 0 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo      M-6 Copy
```

Рис. 2.2: Открытие созданного файла

Вставим в открытый файл код из описания лабораторной работы (Рис. 2.3):

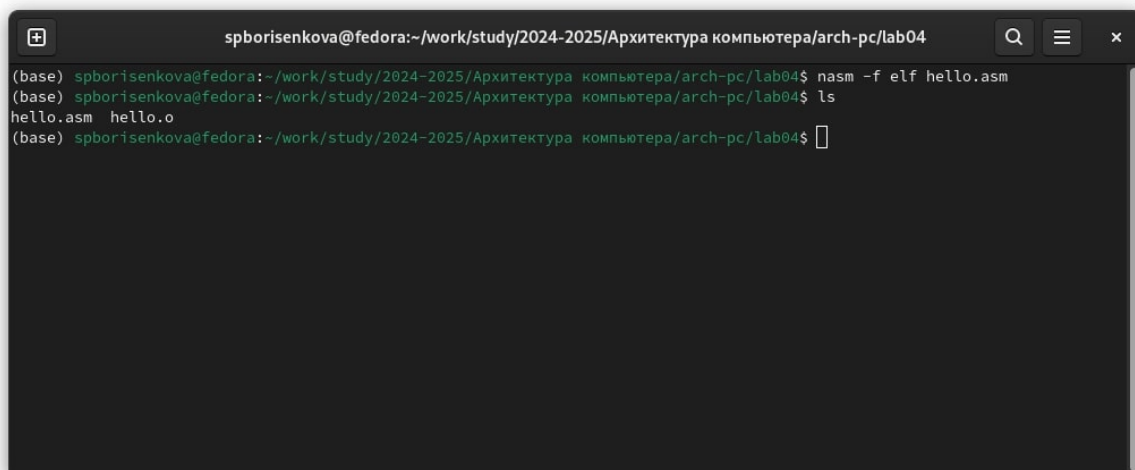


```
GNU nano 7.2 hello.asm Modified
; hello.asm
SECTION .data ; Начало секции данных
hello: DB 'Hello world!',10 ; 'Hello world!' плюс
; символ перевода строки
helloLen: EQU $-hello ; Длина строки hello
SECTION .text ; Начало секции кода
GLOBAL _start
_start: ; Точка входа в программу
mov eax,4 ; Системный 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

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo      M-G Copy
```

Рис. 2.3: Редактирование файла

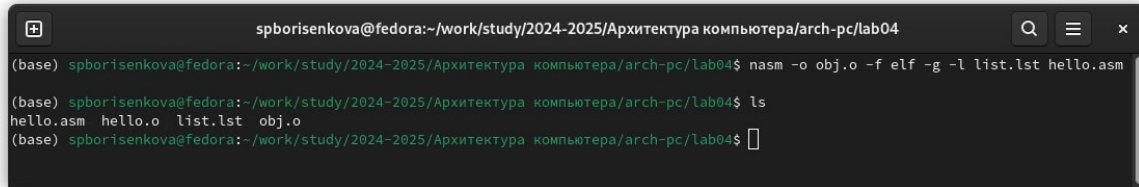
Теперь нам необходимо превратить наш файл в объектный. Этим занимается транслятор NASM. Введём следующую команду и проверим, создался ли объектный файл с помощью команды `ls` (Рис. 2.4):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$ nasm -f elf hello.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$ ls
hello.asm  hello.o
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$
```

Рис. 2.4: Проверка успешного создания файла краткой командой

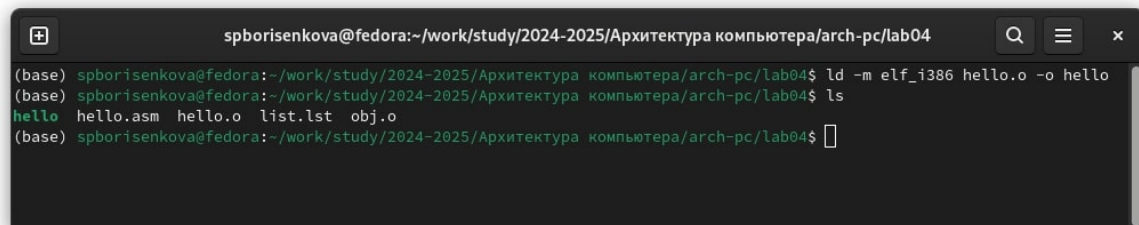
Теперь попробуем использовать полный вариант команды NASM и проверим, создался ли файл с помощью команды `ls` (Рис. 2.5):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$
```

Рис. 2.5: Проверка успешного создания файла полной командой

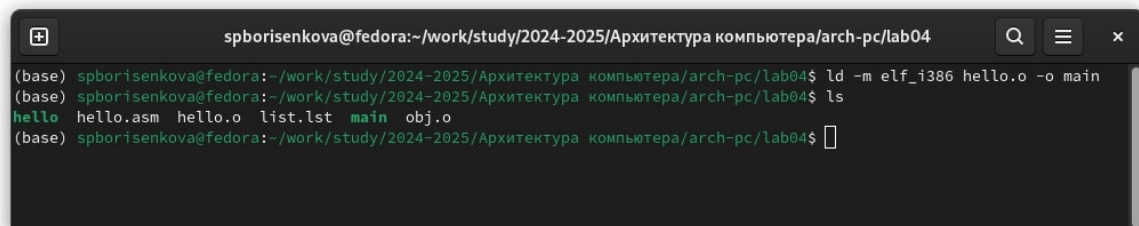
Для создания исполняемого файла необходимо использовать компоновщик ld, который соберёт объектный файл. Введем следующую команду и проверим, создался ли файл с помощью команды ls (Рис. 2.6):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$
```

Рис. 2.6: Проверка на успешное создание исполняемого файла

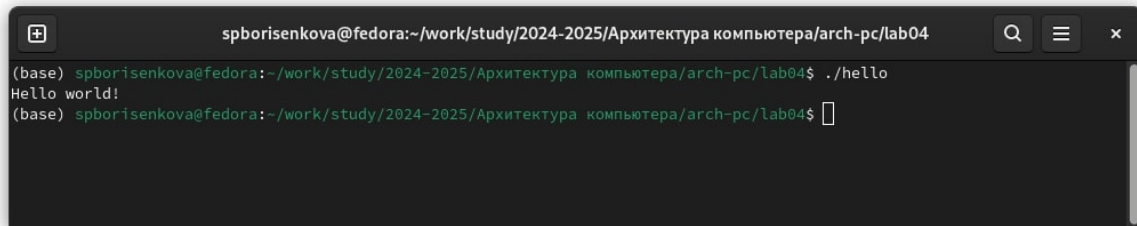
Теперь соберём файл obj.o в файл main и проверим, создался ли файл. (Рис. 2.7):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$ ld -m elf_i386 hello.o -o main
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$
```

Рис. 2.7: Проверка на успешное создание файла

Теперь запустим файл hello.(Рис. 2.8):

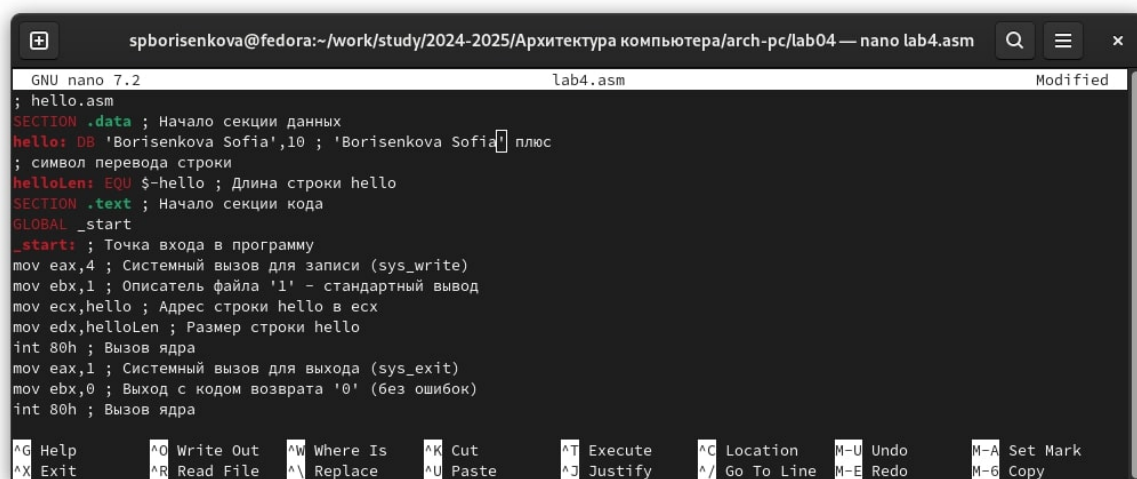
A terminal window with a dark background. The title bar shows the user 'spborisenkova@fedora' and the current directory '/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04'. The terminal contains two lines of text: a command prompt '(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04\$' followed by './hello', and the output 'Hello world!'. A second prompt line is visible below the output.

```
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$ ./hello
Hello world!
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$
```

Рис. 2.8: Запуск исполняемого файла hello

3 Выполнение задания для самостоятельной работы

Скопируем файл `hello.asm` в каталог `~/work/arch-pc/lab04` под названием `lab4.asm` и внесём изменения в скопированный файл. Изменим третью строчку, заменив фразу `Hello world!` на фамилию и имя (Рис. 3.1):



```
GNU nano 7.2 lab4.asm Modified
; hello.asm
SECTION .data ; Начало секции данных
hello: DB 'Borisenkova Sofia',10 ; 'Borisenkova Sofia' 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

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo      M-G Copy
```

Рис. 3.1: Процесс редактирования файла

Теперь скомпилируем полученный файл в объектный, соберём полученный объектный файл. Укажем, что выходной файл должен быть назван `lab4` и запустим собранный файл (Рис. 3.2):

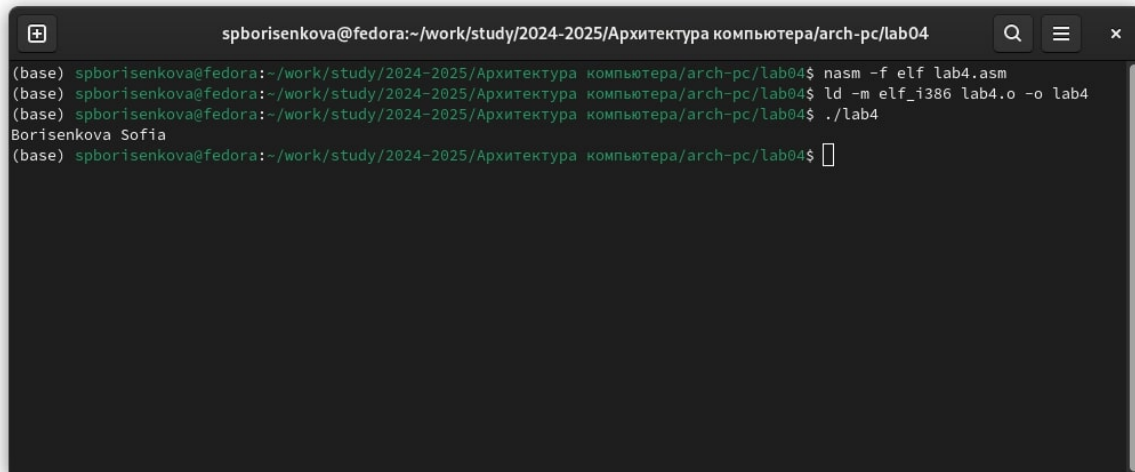
A terminal window with a dark background and light green text. The window title is "spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04". The terminal shows four lines of commands and their outputs: 1. Command: `nasm -f elf lab4.asm`; Output: `(base) spborisenkova@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$`. 2. Command: `ld -m elf_i386 lab4.o -o lab4`; Output: `(base) spborisenkova@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$`. 3. Command: `./lab4`; Output: `(base) spborisenkova@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$` followed by `Borisenkova Sofia` on the next line. 4. The prompt `(base) spborisenkova@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$` is shown again with a cursor. The window has standard Linux window controls (minimize, maximize, close) in the top right corner.

Рис. 3.2: Запуск собранного файла

Теперь скопируем файлы `hello.asm`, `lab4.asm` в каталог 4 лабораторной работы и загрузим результат проделанной лабораторной работы на GitHub.

4 Выводы

В результате выполнения лабораторной работы появилось понимание того, как работает алгоритм создания исполняемого файла из кода на ассемблере, а также появились навыки работы с языком `asm`, компиляции кода в объектный файл и сборкой исполняемых программ.