

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

дисциплина: Архитектура вычислительных систем

Студент: Уханаева Сансара Зоригтуевна

Студ. билет № 1132229047

Группа: НБИбд-03-22

МОСКВА

2022 г.

Цель работы:

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

Выполнение лабораторной работы:

1. Программа *Hello world!*

Рассмотрим пример простой программы на языке ассемблера NASM. Традиционно первая программа выводит приветственное сообщение Hello world! на экран.

Создадим каталог для работы с программами на языке ассемблера NASM и перейдем в созданный каталог:

```
szukhanaeva@dk4n58 ~ $ mkdir ~/work/arch-pc/lab05
szukhanaeva@dk4n58 ~ $ cd ~/work/arch-pc/lab05
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $
```

Создадим текстовый файл с именем hello.asm с помощью команды touch

```
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ touch hello.asm
```

Далее откроем этот файл с помощью текстового редактора mcedit.

```
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ mcedit hello.asm
```

Введем в него следующий текст:

```
; 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
```

```

hello.asm      [----]  0 L:[ 1+ 0  1/ 19] *(0 / 990b) 0059 0x03B
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

```

2. Транслятор NASM.

Для компиляции приведённого выше текста программы «Hello World» напишем:

```
nasm -f elf hello.asm
```

Затем с помощью команды ls проверим, что объектный файл был создан:

```

szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ mcedit hello.asm
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ nasm -f elf hello.asm
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ ls hello.o
hello.o
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $

```

Объектный файл имеет имя «hello.o»

3. Расширенный синтаксис командной строки NASM

Полный вариант командной строки nasm выглядит следующим образом:

```

nasm [-@ косвенный_файл_настроек] [-o объектный_файл] [-f
↪ формат_объектного_файла] [-l листинг] [параметры...] [--]
↪ исходный_файл

```

Выполните следующую команду:

```
nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Выполним следующую команду:

```
nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Данная команда скомпилирует исходный файл hello.asm в obj.o

С помощью команды ls проверьте, что файлы были созданы.

```
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ ls
hello.asm  hello.o  list.lst  obj.o
```

4. Компоновщик LD

Чтобы получить исполняемую программу, объектный файл передадим на обработку компоновщику:

```
ld -m elf_i386 hello.o -o hello
```

```
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ ld -m elf_i386 hello.o -o hello
```

С помощью команды ls проверим, что исполняемый файл hello был создан.

```
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ ld -m elf_i386 hello.o -o hello
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $
```

Выполним следующую команду:

```
ld -m elf_i386 obj.o -o main
```

```
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ ld -m elf_i386 obj.o -o main
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $
```

Имя исполняемый файла - «obj.o»

5. Запуск исполняемого файла

Запустим на выполнение созданный исполняемый файл, находящийся в текущем каталоге, можно, набрав в командной строке:

```
./hello
```

```
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ ./hello
Hello world!
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $
```

Задание для самостоятельной работы

1. В каталоге ~/work/arch-pc/lab05 с помощью команды cp создадим копию файла hello.asm с именем lab5.asm

```
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ cp hello.asm lab5.asm
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $
```

2. С помощью любого текстового редактора внесем изменения в текст программы в файле lab5.asm так, чтобы вместо Hello world! на экран выводилась строка с вашими фамилией и именем.

```
lab5.asm [-M--] 34 L:[ 1+ 2 3/ 20] *(132 /1036b) 0097 0x061
; hello.asm
SECTION .data                                ; Начало секции данных
    hello:    DB 'Ukhanaeva Sansara',10      ; 'Ukhanaeva Sansara' 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
```

3. Оттранслируем полученный текст программы lab5.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл.

```
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ ld -m elf_i386 lab5.o -o lab5
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ ls
hello hello.asm hello.o lab5 lab5.asm lab5.o list.lst main obj.o
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ ld -m elf_i386 obj.o -o main
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $ ./lab5
Ukhanaeva Sansara
szukhanaeva@dk4n58 ~/work/arch-pc/lab05 $
```

4. Скопируем файлы hello.asm и lab5.asm в свой локальный репозиторий в каталог ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab05/. Загрузим файлы на Github

Вывод: Я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.

