

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 8

дисциплина: Архитектура компьютера

Студент: Игнатенкова Виктория Станиславовна

Группа: НММбд-02-24

МОСКВА

2024 г.

СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ.....	3
2. ЗАДАНИЕ.....	4
3. ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ.....	5
4. ВЫВОД.....	6

1. ЦЕЛЬ РАБОТЫ

Освоить практическое применение циклов и обработку аргументов командной строки при написании программ.

2. ЗАДАНИЕ

1. Организация стека:

Написать программу, использующую стек для хранения и извлечения данных.

2. Инструкции организации циклов:

Описать назначение и синтаксис инструкций организации циклов в NASM (с примерами).

3. Реализация циклов в NASM:

Написать программу, вычисляющую факториал числа с использованием цикла.

4. Обработка аргументов командной строки:

Написать программу, выводящую все переданные аргументы командной строки.

5. Задание для самостоятельной работы:

Разработать программу, использующую циклы, стек и обработку аргументов командной строки

3. ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

Создаём каталог для программ лабораторной работы № 8, перейдём в него и создайте файл lab8-1.asm:

```
vsignatenkova@dk3n55 ~ $ mkdir ~/work/arch-pc/lab08
vsignatenkova@dk3n55 ~ $ cd ~/work/arch-pc/lab08
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ touch lab8-1.asm
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ls
lab8-1.asm
```

Рис.3.1. Каталог

Введём в файл lab8-1.asm текст программы из листинга 8.1. Создаём исполняемый файл и проверим его работу:

```
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf -l lab8-1.lst lab8-1.asm
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 3
3
2
1
```

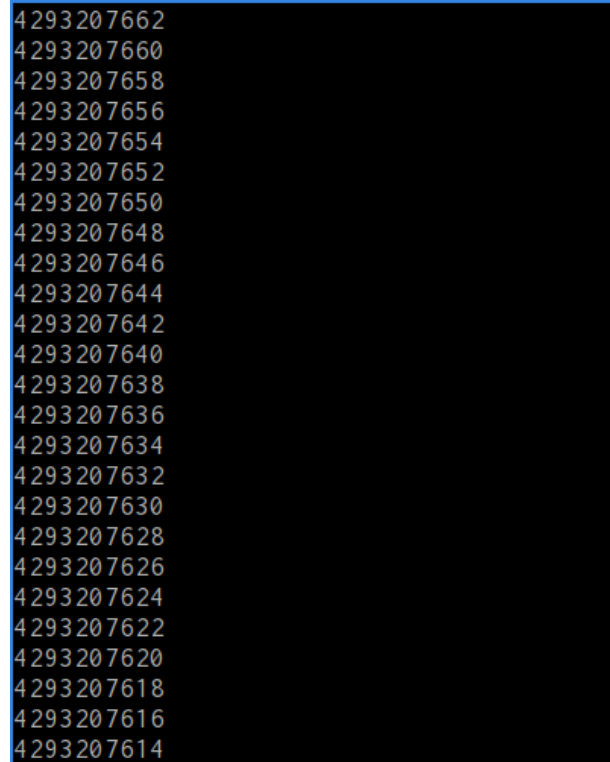
Рис.3.2. Проверка

Вносим изменения:

```
22 mov [N],eax
23 ; ----- Организация цикла
24 mov ecx,[N] ; Счетчик цикла, 'ecx=N'
25 label:
26 label:
27 sub ecx,1 ; 'ecx=ecx-1'
28 mov [N],ecx
29 mov eax,[N]
30 call iprintLF
31 loop label
```

Рис.3.3. Изменения

Проверка:



4293207662
4293207660
4293207658
4293207656
4293207654
4293207652
4293207650
4293207648
4293207646
4293207644
4293207642
4293207640
4293207638
4293207636
4293207634
4293207632
4293207630
4293207628
4293207626
4293207624
4293207622
4293207620
4293207618
4293207616
4293207614

Рис.3.4. Бесконечность

Вносим изменения в текст программы, добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop:

```
24 mov ecx,[N] ; Счетчик цикла, 'ecx=N'
25 label:
26
27 push ecx
28 sub ecx,1
29 mov [N],ecx
30 mov eax,[N]
31 call iprintLF
32 pop ecx
33
34 loop label
35
36 call quit
```

Рис.3.5. Изменения

Создаём исполняемый файл и проверяем его работу:

```
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ gedit lab8-1.asm
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf -l lab8-1.lst lab8-1.asm
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 8
7
6
5
4
3
2
1
0
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $
```

Рис.3.6. Проверка

Создаём файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 и введём в него текст программы из листинга 8.2:

```
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $
```

Рис.3.7. Проверка

Рассмотрим еще один пример программы, которая выводит сумму чисел, которые передаются в программу как аргументы. Создаём файл lab8-3.asm в каталоге ~/work/arch-pc/lab08 и введём в него текст программы из листинга 8.3:

```
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ touch lab8-3.asm
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ gedit lab8-3.asm
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf -l lab8-3.lst lab8-3.asm
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ./main 1 3 2 4 5
bash: ./main: Нет такого файла или каталога
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-3 1 3 2 4 5
Результат: 15
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-3 10 11 12 13 14
Результат: 60
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $
```

Рис.3.8. Проверка

Исправляем программу сложения на операцию умножения:

```
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ gedit lab8-3.asm
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf -l lab8-3.lst lab8-3.asm
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-3 10 11 12 13 14
Результат: 240240
```

Рис.3.9. Проверка

САМОСТОЯТЕЛЬНАЯ РАБОТА

Напишем программу, которая находит сумму значений функций, у меня 15 вариант :

$$f(x)=13+6x$$

```
SECTION .data
prim DB 'f(x)=13 + 6x',0
otv DB 'Результат: ',0
SECTION .text
GLOBAL _start
_start:

pop ecx

pop edx

sub ecx,1

mov esi,0

mov eax,prim
call sprintLF
next:
cmp ecx,0
jz _end

mov ebx,6
pop eax
```

Рис.1. Программа

Создайте исполняемый файл и проверьте его работу на нескольких наборах:

```
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ touch lab8-4.asm
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ gedit lab8-4.asm
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf -l lab8-4.lst lab8-4.asm
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-4 lab8-4.o
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-4 1 2 3 4
f(x)=13 + 6x
Результат: 112
vsignatenkova@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-4 5 6 7 8
f(x)=13 + 6x
Результат: 208
```

Рис.2. Проверка нашей программы

4. ВЫВОД

Успешно освоили практическое применение циклов и обработку аргументов командной строки, что подтверждается выполненными заданиями и демонстрирует их способность создавать функциональные и гибкие программы.