

Лабораторная работа №9

Дисциплина: Архитектура компьютера

Кондратьев Арсений Вячеславович

06.10.2022

Содержание

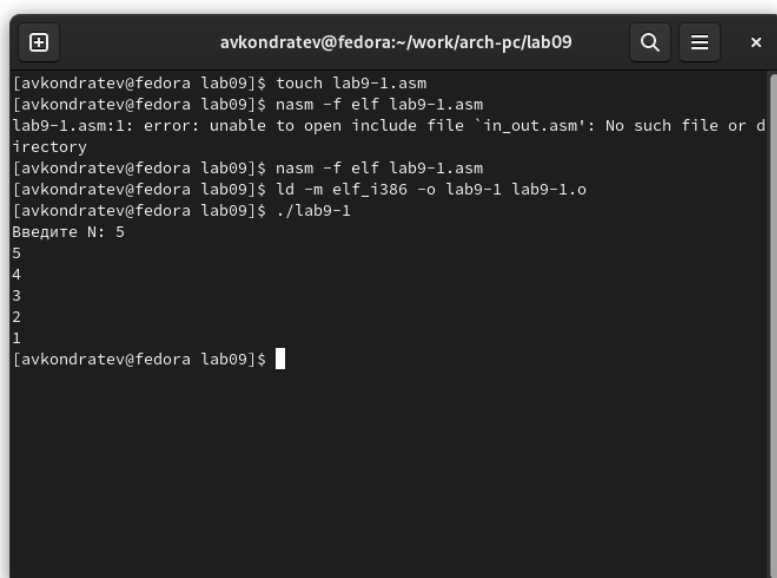
1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Выводы	8
4	Контрольные вопросы	9

1 Цель работы

Приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

2 Выполнение лабораторной работы

1. Выполнил программу вывода значений регистра esx(рис.2.1)



```
avkondratev@fedora:~/work/arch-pc/lab09
[avkondratev@fedora lab09]$ touch lab9-1.asm
[avkondratev@fedora lab09]$ nasm -f elf lab9-1.asm
lab9-1.asm:1: error: unable to open include file `in_out.asm': No such file or directory
[avkondratev@fedora lab09]$ nasm -f elf lab9-1.asm
[avkondratev@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[avkondratev@fedora lab09]$ ./lab9-1
Введите N: 5
5
4
3
2
1
[avkondratev@fedora lab09]$
```

Figure 2.1: Рис. 1

2. Изменил текст программы, добавив изменение значение регистра esx в цикле(рис.2.2)

```
avkondratev@fedora:~/work/arch-pc/lab09
4294907588
4294907586
4294907584
4294907582
4294907580
4294907578
4294907576
4294907574
4294907572
4294907570
4294907568
4294907566
4294907564
4294907562
4294907560
4294907558
4294907556
4294907554
4294907552
4294907550
4294907548
4294907546
42^C
[avkondratev@fedora lab09]$
```

Figure 2.2: Рис. 2

3. Внес изменения в текст программы, добавив команды `push` и `pop` (рис.2.3)

```
avkondratev@fedora:~/work/arch-pc/lab09
4294907570
4294907568
4294907566
4294907564
4294907562
4294907560
4294907558
4294907556
4294907554
4294907552
4294907550
4294907548
4294907546
42^C
[avkondratev@fedora lab09]$ nasm -f elf lab9-1.asm
[avkondratev@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[avkondratev@fedora lab09]$ ./lab9-1
Введите N: 5
4
3
2
1
0
[avkondratev@fedora lab09]$
```

Figure 2.3: Рис. 3

4. Написал программу выводящую на экран аргументы командной строки(рис.2.4)

```
avkondratev@fedora:~/work/arch-pc/lab09
4294907552
4294907550
4294907548
4294907546
42^C
[avkondratev@fedora lab09]$ nasm -f elf lab9-1.asm
[avkondratev@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[avkondratev@fedora lab09]$ ./lab9-1
Введите N: 5
4
3
2
1
0
[avkondratev@fedora lab09]$ nasm -f elf lab9-2.asm
[avkondratev@fedora lab09]$ ld -m elf_i386 -o lab9-2 lab9-2.o
[avkondratev@fedora lab09]$ ./lab9-2
[avkondratev@fedora lab09]$ ./lab9-2
[avkondratev@fedora lab09]$ ./lab9-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
[avkondratev@fedora lab09]$
```

Figure 2.4: Рис. 4

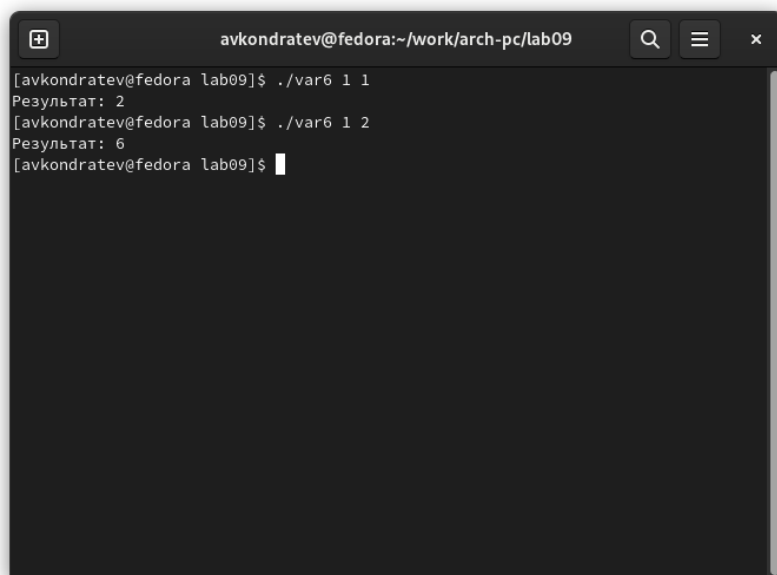
5. Написал программу вычисления суммы аргументов командной строки(рис.2.5)

```
avkondratev@fedora:~/work/arch-pc/lab09
42^C
[avkondratev@fedora lab09]$ nasm -f elf lab9-1.asm
[avkondratev@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[avkondratev@fedora lab09]$ ./lab9-1
Введите N: 5
4
3
2
1
0
[avkondratev@fedora lab09]$ nasm -f elf lab9-2.asm
[avkondratev@fedora lab09]$ ld -m elf_i386 -o lab9-2 lab9-2.o
[avkondratev@fedora lab09]$ ./lab9-2
[avkondratev@fedora lab09]$ ./lab9-2
[avkondratev@fedora lab09]$ ./lab9-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
[avkondratev@fedora lab09]$ nasm -f elf lab9-3.asm
[avkondratev@fedora lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[avkondratev@fedora lab09]$ ./lab9-3 12 13 7 10 5
Результат: 47
[avkondratev@fedora lab09]$
```

Figure 2.5: Рис. 5

6. Напишите программу, которая находит сумму значений функции в соот-

ветствии с 6 вариантом(рис.2.6)

A terminal window with a dark background and light text. The title bar shows the user 'avkondratev' on a 'fedora' machine, in the directory '~/work/arch-pc/lab09'. The terminal contains three lines of text: a command './var6 1 1' followed by its output 'Результат: 2', another command './var6 1 2' followed by its output 'Результат: 6', and a final prompt line with a cursor. The window has standard OS controls (minimize, maximize, close) and a search icon in the top right.

```
avkondratev@fedora:~/work/arch-pc/lab09
[avkondratev@fedora lab09]$ ./var6 1 1
Результат: 2
[avkondratev@fedora lab09]$ ./var6 1 2
Результат: 6
[avkondratev@fedora lab09]$
```

Figure 2.6: Рис. 6

3 Выводы

Я приобрел навыки написания программ с использованием циклов и обработкой аргументов командной строки.

4 Контрольные вопросы

1. Инструкция `loop` выполняется в два этапа. Сначала из регистра `ecx` вычитается единица и его значение сравнивается с нулём. Если регистр не равен нулю, то выполняется переход к указанной метке. Иначе переход не выполняется и управление передаётся команде, которая следует сразу после команды `loop`
2. С помощью условного оператора, который будет возвращать нас в начальную позицию цикла, пока не будет выполнено условие
3. Стек — это структура данных, организованная по принципу LIFO («Last In — First Out» или «последним пришёл — первым ушёл»). Стек является частью архитектуры процессора и реализован на аппаратном уровне
4. Стек имеет вершину, адрес последнего добавленного элемента, который хранится в регистре `esp` (указатель стека). Противоположный конец стека называется дном. Значение, помещённое в стек последним, извлекается первым