

Лабораторная работа №8.
Программирование цикла. Обработка
аргументов командной строки.

НПМбВ-01-21

Ермаков Алексей

Содержание

Цель работы	5
Задание	6
Теоретическое введение	7
Выполнение лабораторной работы	8
Выводы	12

Список таблиц

Список иллюстраций

1	8-1 Создание и проверка работы цикла	8
2	8-2 Пример некорректной работы при изменении значение регистра esx в цикле	9
3	8-3 Работа исправленного файла	9
4	8-4 Пример обработки различного количества аргументов	10
5	8-5 Пример вычисления суммы аргументов командной строки	10
6	8-6 Пример вычисления произведения аргументов командной строки	10
7	8-7 Пример работы программы вычисления суммы $f(x) = 10x-5$	11

Цель работы

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

Задание

Напишите программу, которая находит сумму значений функции $f(x)$ для $x = x_1, x_2, \dots, x_n$, т.е. программа должна выводить значение $f(x_1) + f(x_2) + \dots + f(x_n)$. Значения x_i передаются как аргументы. Вид функции $f(x)$ выбрать из таблицы 8.1 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах

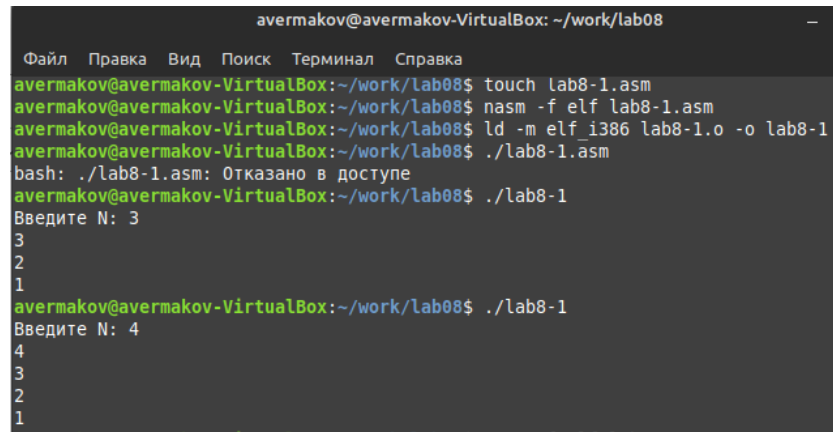
Теоретическое введение

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. Для работы со стеком в процессоре есть специальные регистры (ss, bp, sp) и команды. Основной функцией стека является функция сохранения адресов возврата и передачи аргументов при вызове процедур. Кроме того, в нём выделяется память для локальных переменных и могут временно храниться значения регистров.

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. При помещении значения в стек указатель стека уменьшается, а при извлечении — увеличивается.

Выполнение лабораторной работы

Создал каталог для программ лабораторной работы, написал в файл lab8-1.asm текст программы из листинга 8.1. Создал исполняемый файл и проверил его работу. (рис. @fig:001).



```
avermakov@avermakov-VirtualBox: ~/work/lab08
Файл  Правка  Вид  Поиск  Терминал  Справка
avermakov@avermakov-VirtualBox:~/work/lab08$ touch lab8-1.asm
avermakov@avermakov-VirtualBox:~/work/lab08$ nasm -f elf lab8-1.asm
avermakov@avermakov-VirtualBox:~/work/lab08$ ld -m elf_i386 lab8-1.o -o lab8-1
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-1.asm
bash: ./lab8-1.asm: Отказано в доступе
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-1
Введите N: 3
3
2
1
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-1
Введите N: 4
4
3
2
1
```

Рис. 1: 8-1 Создание и проверка работы цикла

Изменил текст программы добавив изменение значения регистра ехх в цикле (рис. @fig:002).


```

avermakov@avermakov-VirtualBox:~/work/lab08$ touch lab8-1.asm
avermakov@avermakov-VirtualBox:~/work/lab08$ nasm -f elf lab8-1.asm
avermakov@avermakov-VirtualBox:~/work/lab08$ ld -m elf_i386 lab8-1.o -o lab8-1
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-1.asm
bash: ./lab8-1.asm: Отказано в доступе
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-1
Введите N: 3
3
2
1
1
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-1
Введите N: 4
4
3
2
1
avermakov@avermakov-VirtualBox:~/work/lab08$ nasm -f elf lab8-1.asm
avermakov@avermakov-VirtualBox:~/work/lab08$ ld -m elf_i386 lab8-1.o -o lab8-1
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-1
Введите N: 3
2
0
4294967294
4294967292
4294967290
4294967288
4294967286

```

Рис. 2: 8-2 Пример некорректной работы при изменении значение регистра ехх в цикле

Внес изменения в текст программы добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop. Создал исполняемый файл и проверил его работу. (рис. @fig:003).

```

4294362046
4294362044
4294362042
4294362040
4294362038
4294362036
^C
avermakov@avermakov-VirtualBox:~/work/lab08$ nasm -f elf lab8-1.asm
avermakov@avermakov-VirtualBox:~/work/lab08$ ld -m elf_i386 lab8-1.o -o lab8-1
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-1
Введите N: 3
2
avermakov@avermakov-VirtualBox:~/work/lab08$

```

Рис. 3: 8-3 Работа исправленного файла

Создал исполняемый файл и запустил его, указав аргументы: аргумент1 аргумент 2 'аргумент 3'. В данном случае было обработано 4 аргумента, так как часть строки "аргумент 2" разделена пробелом и не имеет '' данных символов, а значит воспринимается как два разных аргумента (рис. @fig:004).

```

avermakov@avermakov-VirtualBox:~/work/lab08$ touch lab8-2.asm
avermakov@avermakov-VirtualBox:~/work/lab08$ nasm -f elf lab8-2.asm
avermakov@avermakov-VirtualBox:~/work/lab08$ ld -m elf_i386 lab8-2.o -o lab8-2
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
2
аргумент 3
avermakov@avermakov-VirtualBox:~/work/lab08$

```

Рис. 4: 8-4 Пример обработки различного количества аргументов

Создал исполняемый файл и запустил его, указав аргументы 12 13 7 10 5 (рис. @fig:005).

```

avermakov@avermakov-VirtualBox:~/work/lab08$ touch lab8-3.asm
avermakov@avermakov-VirtualBox:~/work/lab08$ nasm -f elf lab8-3.asm
avermakov@avermakov-VirtualBox:~/work/lab08$ ld -m elf_i386 lab8-3.o -o lab8-3
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-3 12 13 7 10 5
Результат: 47
avermakov@avermakov-VirtualBox:~/work/lab08$

```

Рис. 5: 8-5 Пример вычисления суммы аргументов командной строки

Изменил текст программы для вычисления произведения аргументов командной строки. Создал исполняемый файл и запустил его, указав разные аргументы. (рис. @fig:006).

```

avermakov@avermakov-VirtualBox:~/work/lab08$ nasm -f elf lab8-3.asm
avermakov@avermakov-VirtualBox:~/work/lab08$ ld -m elf_i386 lab8-3.o -o lab8-3
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-3 1 2 3 4 5
Результат: 120
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-3 12 13 7 10 5
Результат: 54600
avermakov@avermakov-VirtualBox:~/work/lab08$

```

Рис. 6: 8-6 Пример вычисления произведения аргументов командной строки

Написал программу, которая находит сумму значений функции $f(x)$ для $x = x_1, x_2, \dots, x_n$, т.е. программа должна выводить значение $f(x_1) + f(x_2) + \dots + f(x_n)$ ($f(x) = 10x - 5$) (рис. @fig:007).

```
avermakov@avermakov-VirtualBox:~/work/lab08$ touch lab8-4.asm
avermakov@avermakov-VirtualBox:~/work/lab08$ nasm -f elf lab8-4.asm
avermakov@avermakov-VirtualBox:~/work/lab08$ ld -m elf_i386 lab8-4.o -o lab8-4
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-4 1 2 3
f(x) = 10x-5
Результат: 45
avermakov@avermakov-VirtualBox:~/work/lab08$ ./lab8-4 1 2 3 4
f(x) = 10x-5
Результат: 80
avermakov@avermakov-VirtualBox:~/work/lab08$
```

Рис. 7: 8-7 Пример работы программы вычисления суммы $f(x) = 10x-5$

Выводы

Были получены навыки написания программ с использованием циклов и обработкой аргументов командной строки