

Отчёт по лабораторной работе №13

**Средства, применяемые при разработке программного обеспечения в
ОС типа UNIX/Linux**

Киньябаева Аиша Иделевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Контрольные вопросы	11
5	Выводы	13

Список иллюстраций

3.1	1 задание	7
3.2	Результат	8
3.3	2 задание”	8
3.4	Результат	9
3.5	3 задание	10

Список таблиц

1 Цель работы

Целью данной работы является изучение основ программирования в оболочке ОС UNIX, обучение более сложным командным файлам с использованием логических управляющих конструкций и циклов.

2 Задание

Создание командных файлов

3 Выполнение лабораторной работы

Пишу первую программу, которая должна копировать действия команды семафор (рис. 3.1)

```
home > aikinyabaeva > work > os > lab13 > M Makefile
1  CC = gcc
2  CFLAGS =
3  LIBS = -lm
4
5  # линковка в конечный файл
6  calcul: calculate.o main.o
7  | $(CC) calculate.o main.o -o calcul $(LIBS)
8
9  # компиляция calculate.c
10 calculate.o: calculate.c calculate.h
11 | $(CC) -c calculate.c $(CFLAGS)
12
13 # Компиляция main.c
14 main.o: main.c calculate.h
15 | $(CC) -c main.c $(CFLAGS)
16
17 # Очистка от .o файлов в директории
18 clean:
19 | -rm -f calcul *.o *-gdb ./calcul
```

Рис. 3.1: 1 задание

Далее запускаю программу и смотрю на полученный результат (рис. 3.2)

```

home > aikinjyabaeva > work > os > lab13 > C main.c > main(void)
1  #include <stdio.h>
2
3  #include "calculate.h"
4
5  int main(void) {
6      float Numeral;
7      char Operation[4];
8      float Result;
9      printf("Число: ");
10     scanf("%f", &Numeral);
11     printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
12     scanf("%s", Operation);
13     Result = Calculate(Numeral, Operation);
14     printf("%.2f\n", Result);
15     return 0;
16 }

```

Рис. 3.2: Результат

Пишу вторую программу, которая должна осуществлять поиск и вывод справок по командам (рис. 3.3)

```

home > aikinjyabaeva > work > os > lab13 > C calculate.h > ...
1  #ifndef CALCULATE_H_
2  #define CALCULATE_H_
3      float Calculate(float Numeral, char Operation[4]);
4  #endif /*CALCULATE_H_*/

```

Рис. 3.3: 2 задание”

Запускаю программу и смотрю полученный результат, т.е. справка на определенную команду (рис. 3.4)


```

home > aikinjyabaeva > work > os > lab13 > C calculate.c > Calcu
1  #include "calculate.h"
2
3  #include <math.h>
4  #include <stdio.h>
5  #include <string.h>
6
7  float Calculate(float Numeral, char Operation[4]) {
8      float SecondNumeral;
9      if (strcmp(Operation, "+", 1) == 0) {
10         printf("Второе слагаемое: ");
11         scanf("%f", &SecondNumeral);
12         return (Numeral + SecondNumeral);
13     } else if (strcmp(Operation, "-", 1) == 0) {
14         printf("Вычитаемое: ");
15         scanf("%f", &SecondNumeral);
16         return (Numeral - SecondNumeral);
17     } else if (strcmp(Operation, "*", 1) == 0) {
18         printf("Множитель: ");
19         scanf("%f", &SecondNumeral);
20         return (Numeral * SecondNumeral);
21     } else if (strcmp(Operation, "/", 1) == 0) {
22         printf("Делитель: ");
23         scanf("%f", &SecondNumeral);
24         if (SecondNumeral == 0) {
25             printf("Ошибка: деление на ноль! ");
26             return (HUGE_VAL);
27         } else
28             return (Numeral / SecondNumeral);
29     } else if (strcmp(Operation, "pow", 3) == 0) {
30         printf("Степень: ");
31         scanf("%f", &SecondNumeral);
32         return (pow(Numeral, SecondNumeral));
33     } else if (strcmp(Operation, "sqrt", 4) == 0)
34         return (sqrt(Numeral));
35     else if (strcmp(Operation, "sin", 3) == 0)
36         return (sin(Numeral));
37     else if (strcmp(Operation, "cos", 3) == 0)
38         return (cos(Numeral));
39     else if (strcmp(Operation, "tan", 3) == 0)
40         return (tan(Numeral));
41     else {
42         printf("Неправильно введено действие ");
43         return (HUGE_VAL);
44     }
45 }

```

Рис. 3.4: Результат

Следующая программа выводит случайную последовательность латинских букв (у меня это последовательность из 10 символов) (рис. 3.5), (рис. ??)

```
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/os/lab_prog$ gcc -c calculate.c
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/os/lab_prog$ gcc -c main.c
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/os/lab_prog$ gcc calculate.o main.o -o calcul -lm
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/os/lab_prog$ ./calcul
Число: 13
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): sqrt
3.61
aikinjyabaeva@aikinjyabaeva-VirtualBox:~/work/os/lab_prog$
```

Рис. 3.5: 3 задание

Результат

Ну и в завершение работы выкладываю все на гит

4 Контрольные вопросы

1. Ошибка в этой строке заключается в использовании квадратных скобок. В командной оболочке Bash они используются для выполнения условных выражений вместе с командой `test`. Правильно будет так: `while (($1 != "exit"))`
2. В bash для объединения нескольких строк в одну можно использовать оператор конкатенации `+` или `..`. Например: `string1="Hello" string2="world" result=$string1$string2 echo $result`

Вывод будет: Helloworld

3. Утилита `seq` в Linux используется для создания итераторов, которые генерируют последовательности чисел. Она имеет следующий синтаксис: `seq ОПЦИИ... ПОСЛЕДНЕЕ`. Например, команда `"seq 1 5"` выведет последовательность чисел от 1 до 5. Некоторые из опций, которые можно использовать с утилитой `seq`: `-f --format=ФОРМАТ` используйте указанный формат для каждого числа. `-s --separator=СЕПАРАТОР` используйте указанный разделитель между числами
4. Результат вычисления выражения будет равен 3, так как при использовании символов двойных круглых скобок происходит округление до ближайшего целого числа.
5. Некоторые отличия командной оболочки `zsh` от `bash` включают:

- Мощная автодополнение и расширенная подсказка, которые делают работу с командами более эффективной и быстрой.
- Zsh имеет более интуитивный синтаксис для управления массивами и строками, что делает ее удобным выбором для обработки текста.
- В zsh доступен “глубокий патчинг”, позволяющий изменять внутреннюю логику самой оболочки.
- Zsh имеет более развитую систему конфигурации и управления плагинами, что делает ее наиболее гибкой и настраиваемой оболочкой в мире Linux.

6. Синтаксис верный, но надо уточнить значение переменной LIMIT. Также, важно помнить, что в данном цикле нет тела цикла - то есть ничего не будет происходить, пока не будет добавлен код, который будет выполняться внутри цикла.

7. C++:

ПРЕИМУЩЕСТВА BASH: - Встроенная поддержка системных команд и утилит - Незаменим при работе с системными скриптами на Unix-подобных операционных системах - Гибкость и легкость в написании коротких скриптов, относительно C++

НЕДОСТАТКИ BASH: - Недостаточная мощность в сравнении с C++ - Ограниченный функционал - Сложные программы сложнее считывать и понимать в bash, чем в C++

5 Выводы

В ходе данной лабораторной работы были изучены основы программирования и созданы командные файлы.