

Структура научной презентации

Презентация лабораторной работы № 13

Ахмад Мд Шешир.

Российский университет дружбы народов, Москва, Россия

..... {.columns align=center} ::: {.column width="70%"}

- Ахмад Мд Шешир
- Студент
- Российский университет дружбы народов

::: ::: {.column width="30%"}

Цель работы

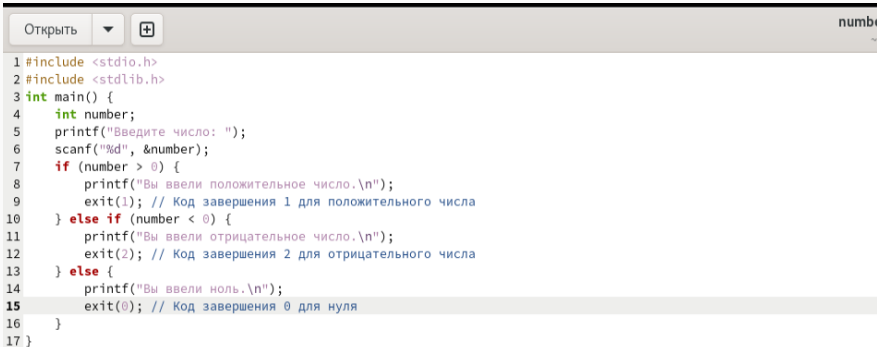
Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

1. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды “\$?”, выдать сообщение о том, какое число было введено.

Открываю текстовый редактор и создаю новый файл для кода на C

```
(base) [sheshir.m@fedora lab13]$ touch number_check.c  
(base) [sheshir.m@fedora lab13]$ gedit number_check.c  
(base) [sheshir.m@fedora lab13]$
```

Программа запрашивает у пользователя ввод числа, определяет является ли оно положительным отрицательным или равным нулю, а затем завершает работу.



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4     int number;
5     printf("Введите число: ");
6     scanf("%d", &number);
7     if (number > 0) {
8         printf("Вы ввели положительное число.\n");
9         exit(1); // Код завершения 1 для положительного числа
10    } else if (number < 0) {
11        printf("Вы ввели отрицательное число.\n");
12        exit(2); // Код завершения 2 для отрицательного числа
13    } else {
14        printf("Вы ввели ноль.\n");
15        exit(0); // Код завершения 0 для нуля
16    }
17 }
```

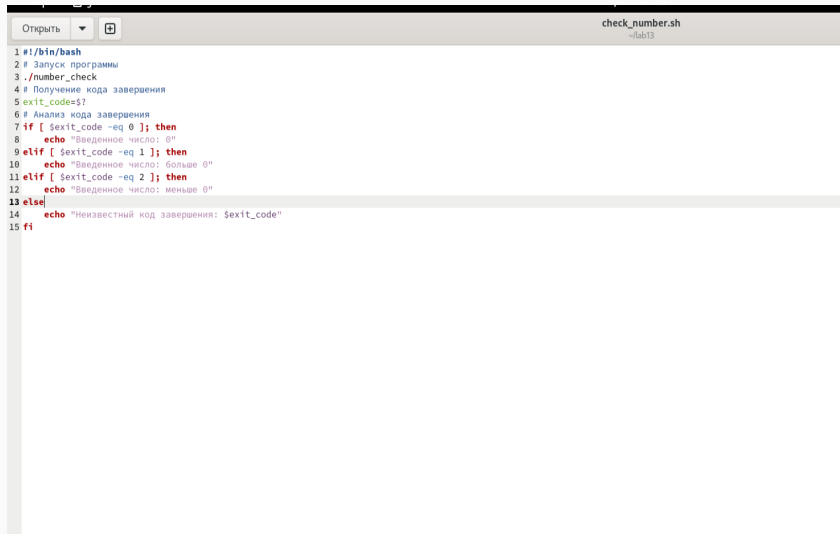
Создаю исполняемый файл, выполнив команду в терминале gcc для языка C

A terminal window with a dark background. The prompt is (base) [sheshir.m@fedora lab13]\$. The command gcc number_check.c -o number_check has been entered and executed. The prompt is now (base) [sheshir.m@fedora lab13]\$ with a white cursor at the end.

```
(base) [sheshir.m@fedora lab13]$ gcc number_check.c -o number_check
(base) [sheshir.m@fedora lab13]$
```

Рис. 2: emacs

Создаю новый файл для кода для BASH и набираю программу



```
1 #!/bin/bash
2 # Запуск программы
3 ./number_check
4 # Получение кода завершения
5 exit_code=$?
6 # Анализ кода завершения
7 if [ $exit_code -eq 0 ]; then
8     echo "Введенное число: 0"
9 elif [ $exit_code -eq 1 ]; then
10    echo "Введенное число: больше 0"
11 elif [ $exit_code -eq 2 ]; then
12    echo "Введенное число: меньше 0"
13 else
14    echo "Неизвестный код завершения: $exit_code"
15 fi
```

Делаю файл исполняемым, выполнив команду в терминале `chmod`

```
(base) [sheshir.m@fedora lab13]$ chmod +x check_number.sh  
(base) [sheshir.m@fedora lab13]$
```

Запускаю исполняемый файл, и пишу любое число 5, все работает

```
(base) [sheshir.m@fedora lab13]$ ./check_number.sh
Введите число: 5
Вы ввели положительное число.
Введенное число: больше 0
(base) [sheshir.m@fedora lab13]$
```

В ходе работы я изучил основы программирования в оболочке ОС UNIX/Linux. Научился писать небольшие командные файлы.