

Лабораторная работа №13

Операционные системы

Краснова К. Г.

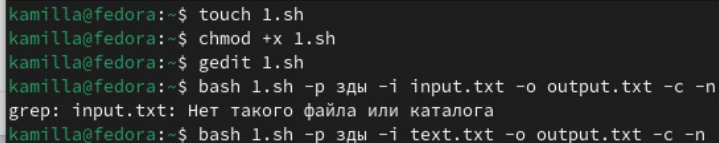
10 мая 2025

Российский университет дружбы народов, Москва, Россия

Цель работы - Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами: `-i` — прочитать данные из указанного файла; `-o` — вывести данные в указанный файл; `-r` — шаблон — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-r`.
2. 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 код завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до `N` (например `1.tmp`, `2.tmp`, `3.tmp`, `4.tmp` и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).

Создаю файл, который будет анализировать командную строку с ключами (рис. 1).

A terminal window with a dark background and light green text. The user 'kamilla' is at a 'fedora' machine in the home directory. They create a file '1.sh' with 'touch', make it executable with 'chmod +x', and open it with 'gedit'. Then they run 'bash 1.sh -p зды -i input.txt -o output.txt -c -n', which results in an error message from 'grep' because 'input.txt' does not exist. Finally, they run the same command with 'text.txt' instead of 'input.txt'.

```
kamilla@fedora:~$ touch 1.sh
kamilla@fedora:~$ chmod +x 1.sh
kamilla@fedora:~$ gedit 1.sh
kamilla@fedora:~$ bash 1.sh -p зды -i input.txt -o output.txt -c -n
grep: input.txt: Нет такого файла или каталога
kamilla@fedora:~$ bash 1.sh -p зды -i text.txt -o output.txt -c -n
```

Рис. 1: Создание файла

Выполнение лабораторной работы

Пример кода, который будет анализировать командную строку (рис. 2).

```
1 #! /bin/bash
2
3 while getopts i:o:p:cn optletter
4 do
5     case $optletter in
6         i) iflag=1; ival=$OPTARG;;
7         o) oflag=1; oval=$OPTARG;;
8         p) pflag=1; pval=$OPTARG;;
9         c) cflag=1;;
10        n) nflag=1;;
11        *) echo Illegal option $optletter;;
12    esac
13 done
14
15 if ! test $cflag
16 then
17     cf=-i
18 fi
19
20 if test $nflag
21 then
22     nf=-n
23 fi
```

Результат выполнения команды (рис. 3).

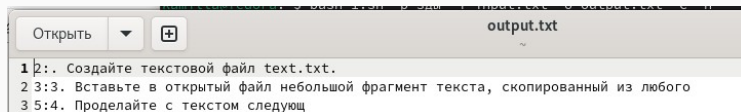
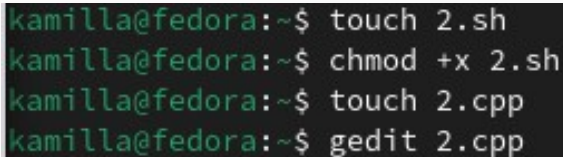


Рис. 3: Результат

Создаю файл для программы на языке C, которая будет определять, является ли число больше нуля (рис. 4).

A terminal window with a dark background and green text. It shows four lines of commands being executed in a shell. The prompt is 'kamilla@fedora:~\$'. The commands are: 'touch 2.sh', 'chmod +x 2.sh', 'touch 2.cpp', and 'gedit 2.cpp'.

```
kamilla@fedora:~$ touch 2.sh
kamilla@fedora:~$ chmod +x 2.sh
kamilla@fedora:~$ touch 2.cpp
kamilla@fedora:~$ gedit 2.cpp
```

Рис. 4: Создание файла

Выполнение лабораторной работы

Пример кода для этого пункта на Си (рис. 5).

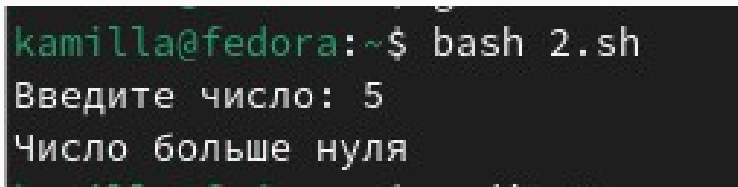
```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main ()
5     int n;
6     printf ("Введите число: ");
7     scanf ("%d", &n);
8     if(n>0){
9         exit(1);
10    }
11    else if (n==0) {
12        exit(0);
13    }
14    else {
15        exit(2);
16    }
```


Код, анализирующий (рис. 6).

```
10 1 #!/bin/bash  
2  
3 gcc -o cprog |2.c  
4 ./cprog  
5 case $? in  
6 0) echo "Число равно нулю";;  
7 1) echo "Число больше нуля";;  
8 2) echo "Число меньше нуля";;  
9 esac  
10
```

Рис. 6: код

Результат выполнения программы (рис. 7).

A terminal window with a black background and green text. The prompt is 'kamilla@fedora:~\$'. The user has entered 'bash 2.sh'. The program outputs 'Введите число: 5' and then 'Число больше нуля'.

```
kamilla@fedora:~$ bash 2.sh
Введите число: 5
Число больше нуля
```

Рис. 7: результат

Выполнение лабораторной работы

Создаю файл для третьего пункта, пишу программу, которая будет создавать и удалять файлы (рис. 8).

```
1 #!/bin/bash  
2 for((i=1; i<=$*; i++))  
3 do  
4 if test -f "$i".tmp  
5 then rm "$i".tmp  
6 else touch "$i.tmp"  
7 fi  
8 done
```

Рис. 8: код

Выполнение лабораторной работы

Результат выполнения (рис. 9).

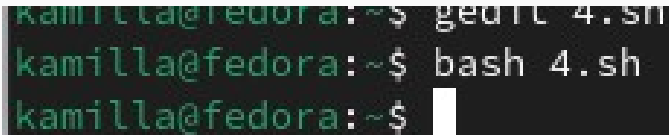
```
kamilla@fedora:~$ bash 3.sh 4
kamilla@fedora:~$ ls
1.sh      Arseny      file.txt
1.tmp     australia   fun
2.c       backup      git-extens
2.cpp     bin         image
2.sh     conf.txt    image.zip
2.tmp     cprog      _index.md
3.sh     Documents  '#lab07.sh
3.tmp     doklad     lab07.sh
4.tmp     Downloads  lab07.sh~
abc1     feathers   LICENSE
kamilla@fedora:~$ bash 3.sh 4
kamilla@fedora:~$ ls
```

Создаю файл для последней программы и вписываю в него код (рис. 10).

```
1 #!/bin/bash  
2 find $* -mtime -7 -mtime +0 -type f > FILES.txt  
3 tar -cf archive.tar -T FILES.txt
```

Рис. 10: код

Результат программы - архив файлов (рис. 11).

A terminal window with a black background and green text. It shows three lines of command execution. The first line is 'kamilla@fedora:~\$ gcc 4.sh'. The second line is 'kamilla@fedora:~\$ bash 4.sh'. The third line is 'kamilla@fedora:~\$' followed by a white cursor block.

```
kamilla@fedora:~$ gcc 4.sh  
kamilla@fedora:~$ bash 4.sh  
kamilla@fedora:~$
```

Рис. 11: результат

В ходе выполнения данной лабораторной работы я: Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.