

Лабораторная работа №11

Операционные системы

Канева Е. П.

22 апреля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Канева Екатерина Павловна
- Студентка группы НКАбд-02-22
- Российский университет дружбы народов
- https://github.com/Nevseros/study_2022-2023_os-intro

Вводная часть

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, а затем ищет в указанном файле нужные строки, определяемые ключом `-p`.
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно большим нуля, меньшим нуля или равным нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку.
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.). Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).
4. Написать командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад.

Выполнение работы

Первое задание

Создала программу, требуемую заданием 1 (рис. 1) и проверила её работу (рис. 2 и 3):

```
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 $optletter is illegal option
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
24
25    grep $cf $nf $pval $ival >> $oval
```

Figure 1: Программа 1.

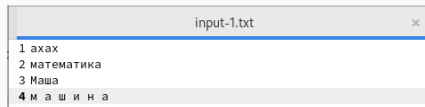


Figure 2: Файл input.

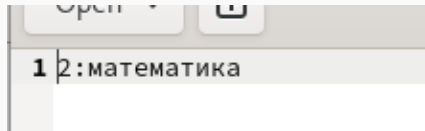


Figure 3: Файл output (вёлся поиск “ма”).

Второе задание

Создала программы, требуемые заданием 2 (рис. 4 и 5) и проверила их работу (рис. 6):

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main() {
5     int n;
6
7     printf("Input a whole number: ");
8     scanf("%d", &n);
9
10    if (n == 0) {
11        exit(0);
12    } else if (n > 0) {
13        exit(1);
14    } else {
15        exit(2);
16    }
17 }
```

Figure 4: Программа 2 (.c).

```
2.sh
1 #!/bin/bash
2
3 gcc -o 2 2.c
4 ./2
5
6 case $? in
7     0) echo "The number equals 0.>";
8     1) echo "The number is above 0.>";
9     2) echo "The number is below 0.>";
10 esac
```

Figure 5: Программа 2 (.sh).

```
[epkaneva@epkaneva ~]$ touch 2.sh
[2]+  Done                  gedit 1.sh
[epkaneva@epkaneva ~]$ touch 2.c
[epkaneva@epkaneva ~]$ gedit 2.sh 2.c &
[1] 3393
[epkaneva@epkaneva ~]$ chmod +x 2.sh
[epkaneva@epkaneva ~]$ bash 2.sh
Input a whole number: 4
The number is above 0.
[epkaneva@epkaneva ~]$ bash 2.sh
Input a whole number: 0
The number equals 0.
[epkaneva@epkaneva ~]$ bash 2.sh
Input a whole number: -123
[epkaneva@epkaneva ~]$ bash 2.sh
Input a whole number: -123
The number is below 0.
```

Figure 6: Запуск программы.

Третье задание

Создала программу, требуемую заданием 3 (рис. 7) и проверила её работу (рис. 8):

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

Figure 7: Программа 3.

```
[epkaneva@epkaneva ~]$ bash 3.sh 3
[epkaneva@epkaneva ~]$ ls
1.sh  2    2.sh  3.sh  4.sh  bin      Documents  input-1.txt  output.txt  Public  Videos
1.tmp 2.c  2.tmp 3.tmp backup Desktop Downloads  Music      Pictures  Templates work
[epkaneva@epkaneva ~]$ bash 3.sh 3
[epkaneva@epkaneva ~]$ ls
1.sh 2.c  3.sh backup Desktop Downloads  Music      Pictures  Templates work
2    2.sh 4.sh bin   Documents input-1.txt output.txt Public  Videos
```

Figure 8: Запуск программы.

Создала программу, требуемую заданием 4 (рис. 9) и проверила её работу (рис. ?? и 10):

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

Figure 9: Программа 4.

```
1 | /home/epkaneva/1.sh  
2 /home/epkaneva/2.sh  
3 /home/epkaneva/2.c  
4 /home/epkaneva/3.sh  
5 /home/epkaneva/4.sh  
6 /home/epkaneva/input-1.txt  
7 /home/epkaneva/output.txt
```

Figure 10: Файл output (поиск в домашней директории).

Заключение

Изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.