

Отчет по лабораторной работе №10

Дисциплина: Операционные системы

Студент: Чепелев Алексей Николаевич

Группа: НПМбв-02-20

Москва 2024 г.

Цель работы

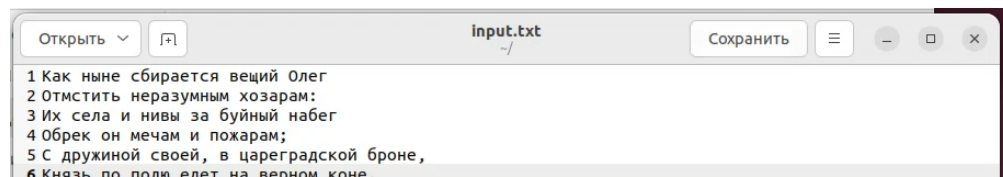
Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Выполнение работы

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку.
2. Создам входящий файл с текстом

```
chepelevan@chepelevan-virtual-machine:~$ qedit input.txt
```

3. Заполняю его



4. Создам командный файл



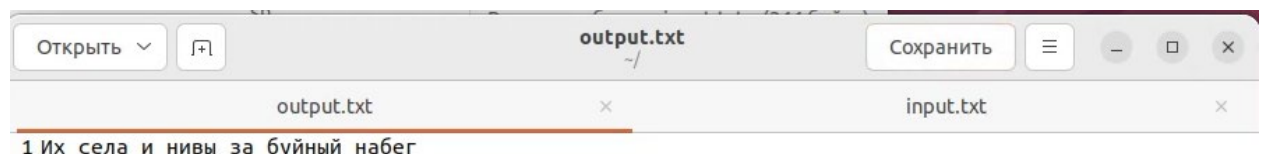
5. Выдаю ему права

```
chepelevan@chepelevan-virtual-machine:~$ chmod +x prog10 1.sh
```

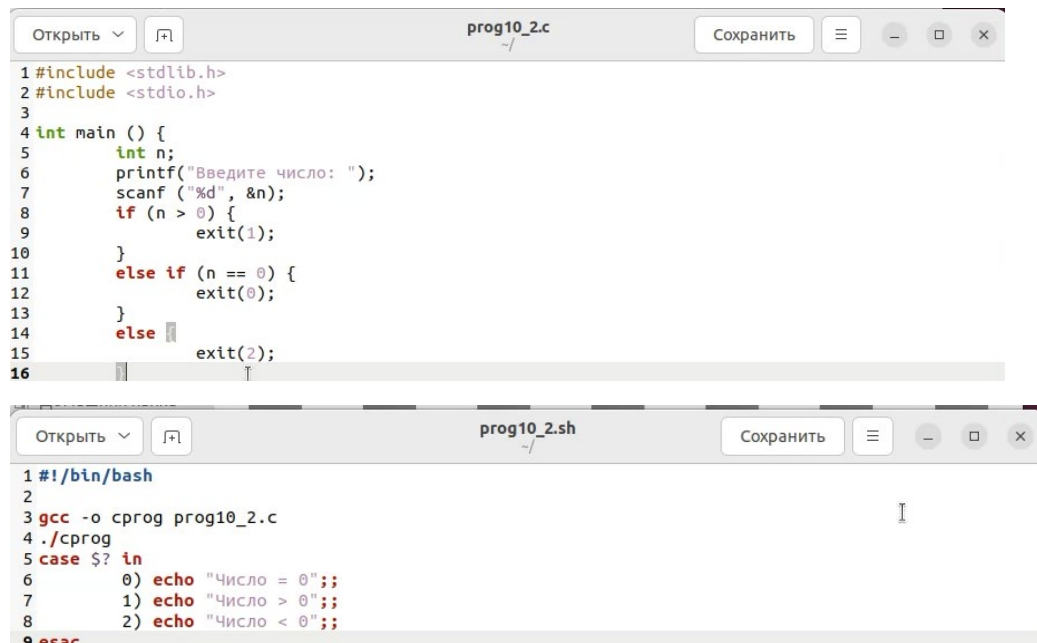
6. Выполняю его

```
chepelevan@chepelevan-virtual-machine:~$ bash prog10 1.sh -p Их -i input.txt -o output.txt -c -n
```

7. Результат:



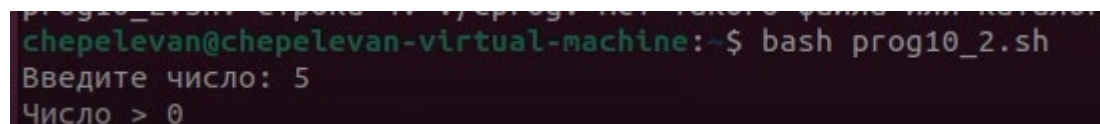
8. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.



```
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    }
```

```
1 #!/bin/bash
2
3 gcc -o cprog prog10_2.c
4 ./cprog
5 case $? in
6     0) echo "Число = 0";;
7     1) echo "Число > 0";;
8     2) echo "Число < 0";;
9 esac
```

9. Выполняю скрипт



```
chepelevan@chepelevan-virtual-machine:~$ bash prog10_2.sh
Введите число: 5
Число > 0
```

10. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N.



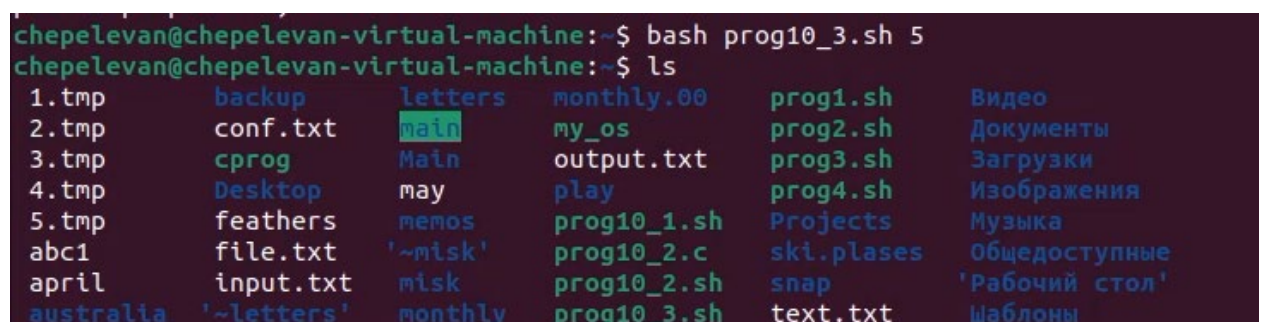
```
1 #!/bin/bash
2
3 for ((i=1; i <= 5; i++))
4 do
5     if test -f "$i".tmp
6     then rm "$i".tmp
7     else touch "$i".tmp
8     fi
9 done
```

11. Выдаю ему права



```
chepelevan@chepelevan-virtual-machine:~$ chmod +x prog10_3.sh
```

12. Выполняю его



```
chepelevan@chepelevan-virtual-machine:~$ bash prog10_3.sh 5
chepelevan@chepelevan-virtual-machine:~$ ls
1.tmp      backup      letters     monthly.00  prog1.sh    Видео
2.tmp      conf.txt    main        my_os       prog2.sh    Документы
3.tmp      cprog      Main        output.txt  prog3.sh    Загрузки
4.tmp      Desktop    may         play        prog4.sh    Изображения
5.tmp      feathers   memos       prog10_1.sh Projects     Музыка
abc1       file.txt   '~misk'     prog10_2.c  ski.plases  Общедоступные
april      input.txt  misk        prog10_2.sh snap        'Рабочий стол'
australia  '~letters' monthly     prog10_3.sh text.txt     Шаблоны
```

13. Написать командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории.

```
Открыть  [икона] *prog10_4.sh Сохранить [меню] [минус] [квадрат] [крестик]
1 #!/bin/bash
2
3 find $* -mtime -7 -mtime +0 -type f > files.txt
4 tar -cf archive.tar -T Files.txt
```

14. Выполняю его

```
chepelevan@chepelevan-virtual-machine:~$ bash prog10_4.sh

chepelevan@chepelevan-virtual-machine:~$ ls
2.tmp      cprog      may         prog10_2.c  snap
3.tmp      Desktop    memos       prog10_2.sh text.txt
4.tmp      feathers   '~misk'     prog10_3.sh Видео
5.tmp      files.txt  misk        prog10_4.sh Документы
abc1       file.txt   monthly     prog1.sh    Загрузки
april      input.txt  monthly.00  prog2.sh    Изображения
archive.tar '~letters' my_os       prog3.sh    Музыка
australia  letters    output.txt  prog4.sh    Общедоступные
backup     main       play        Projects    'Рабочий стол'
```

Вывод:

В этой лабораторной работе я изучил основы программирования в оболочке ОС UNIX.

Ответы на вопросы:

1. getoptс используется для обработки аргументов командной строки. Она позволяет извлекать опции и их значения из списка аргументов.
2. Метасимволы используются в генерации имён файлов для сопоставления шаблонов. Например, звездочка (*) сопоставляет любое количество символов, а знак вопроса (?) сопоставляет любой один символ.
3. Операторы управления действиями используются для изменения потока выполнения скрипта
4. **continue**: Переходит к следующей итерации цикла, пропуская оставшиеся операторы в текущей итерации; **break**: Немедленно выходит из цикла.
5. Команды **false** и **true** используются для возврата кода выхода, указывающего на успех (**true**) или неудачу (**false**).
6. Эта строка проверяет, существует ли файл с именем **man\$\$/\$i.\$\$**. Если файл существует, выполняется оператор **then**.
7. **while**: Выполняет код, пока условие истинно; **until**: Выполняет код, пока условие ложно.