

# Лабораторная работа № 11.

## Программирование в командном процессоре ОС UNIX. Ветвления и циклы

### Цель работы

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

### Последовательность выполнения работы

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами: `-iinputfile` — прочитать данные из указанного файла; `-ooutputfile` — вывести данные в указанный файл; `-rшаблон` — 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 о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.

```
1 //stdio.h>
2 //stdlib.h>
3
4 int main() {
5     int number;
6
7     printf("Введите число: ");
8     scanf("%d", &number);
9
10    if (number > 0) {
11        printf("Число больше нуля\n");
12    } else if (number < 0) {
13        printf("Число меньше нуля\n");
14    } else {
15        printf("Число равно нулю\n");
16    }
17    exit(number);
18 }
```

```

ex=$?
case $ex in
1)
6 echo "Число больше нуля";;
7 2)
8 echo "Число меньше нуля";;
9 0)
echo "Число ноль";;
esac
echo "Код завершения $ex"

```

4. 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, передается в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).

```

in
delete)
num = $2
for ((i=1; i<=num; i++)) do
filename = "${i}.tmp"
touch "$filename"
done
delete)
count=$(ls *tmp | wc -w)
for ((a=0; a<=count; a++))
do
rm "${a}.tmp"
echo "Удаление ${a}.tmp"
done
esac

```

5. Написать командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find).

```

count=$(wc -l < "$TEMP_FILE_LIST")
найденно $FILE_COUNT файлов для архивации."

Создаем архив из найденных файлов
echo "Создание архива..."
4 tar -czf "$ARCHIVE_NAME" -C "$TARGET_DIR" --files-from="$TEMP_FILE_LIST"
15
16 # Очищаем временный файл
7 rm -f "$TEMP_FILE_LIST"

if [ $? -eq 0 ]; then
echo "Архив успешно создан: $ARCHIVE_NAME"
echo "Размер архива: $(du -h "$ARCHIVE_NAME" | cut -f1)"
echo "Количество файлов в архиве: $FILE_COUNT"

echo "Число созданных архивов!"

```

```

./13.csh /home/osipovna/lab13 recent_files
...файлов, измененных менее недели назад в '/home/osipovna/lab13
...дано 5 файлов для архивации.
Создание архива...
tar: Удаляется начальный '/' из имен объектов
tar: Удаляются начальные '/' из цепей жестких ссылок
Архив успешно создан: recent_files.tar.gz
Размер архива: 4,8K
Количество файлов в архиве: 5

```