Отчет по лабораторной работе №10(11)

дисциплина: "Операционные системы"

Студент: Рыскалова Екатерина Андреевна

Группа: НПМбв02-20

Цель работы

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

Ход 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; – -pшаблон — указать шаблон для поиска; – -C — различать большие и малые буквы; – -n — выдавать номера строк.

```
#!/bin/bash

while getopts :i:o:p:cn opt
do
        case "${opt}" in
                i)input=${OPTARG};;
                o)output=${OPTARG};;
                p)mask=${OPTARG};;
                c)c=true;;
                n)n=true;;
        esac
done

if [ $c ]
then
        if [ $n ]
        then
                grep-n -i $mask $input > $output
                exit 0
        else
                grep -i $mask $input > $output
                exit 0
        fi
else
        grep -n $mask $input > $output
        exit 0

fi
~
~
~
~
```

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.

```
[eryskalova@earyskalova lab11]$ cd
[eryskalova@earyskalova ~]$ emacs lab11_2
MESA: error: ZINK: failed to choose pdev
glx: failed to create drisw screen
failed to load driver: zink
[eryskalova@earyskalova ~]$
```

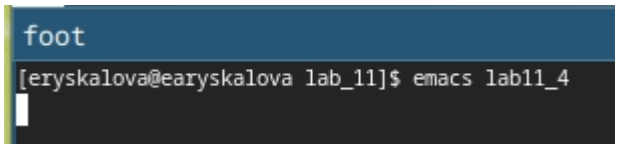
```
lab11_2 - GNU Emacs at earyskalova
File  Edit  Options  Buffers  Tools  Help
Save  Undo
echo "Insert num"
read n
if [ $n -gt 0 ]
then echo ">0"
elif [ $n -eq 0 ]
then echo "=0"
else echo "<0"
fi
```

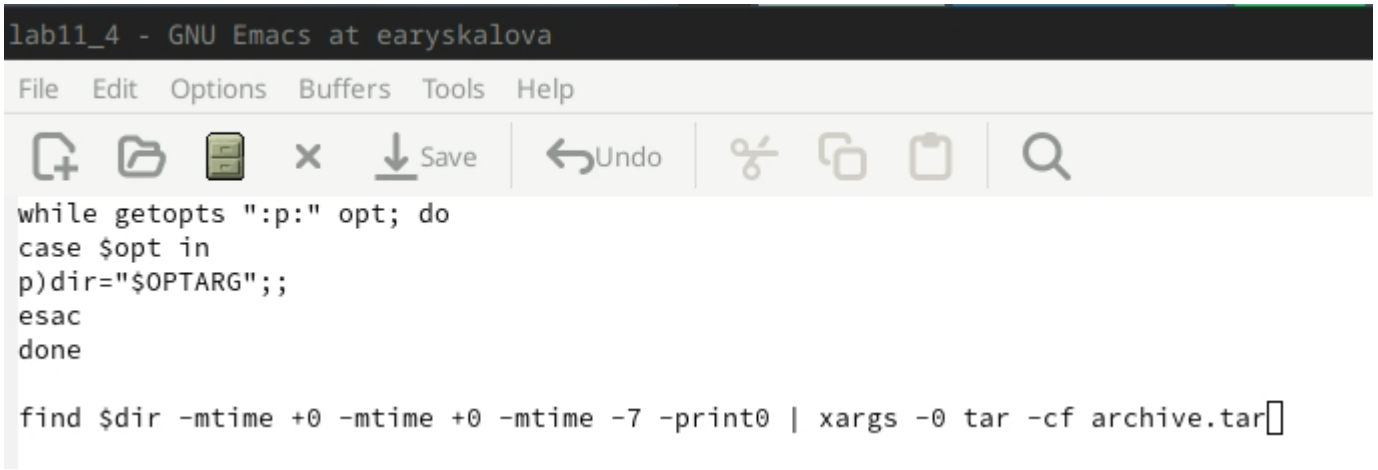
```
[eryskalova@earyskalova ~]$ chmod +x lab11_2
[eryskalova@earyskalova ~]$ ./lab11_2
Insert num
4
>0
[eryskalova@earyskalova ~]$ ./lab11_2
Insert num
0
=0
[eryskalova@earyskalova ~]$
```

3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до □ (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).

```
[eryskalova@earyskalova ~]$ emacs lab11_3
```

4. 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).





Контрольные вопросы

1. Каково предназначение команды getopts?

Команда `getopts` используется для обработки позиционных параметров командной строки (опций и их аргументов) внутри скриптов на Bash. Она позволяет легко разбирать опции командной строки, переданные скрипту, и обрабатывать их, упрощая написание скриптов.

2. Какое отношение метасимволы имеют к генерации имён файлов?

Метасимволы (глобальные символы) в оболочке Unix, такие как `*`, `?` и `[...]`, используются для генерации имен файлов с помощью шаблонов. Эти символы позволяют сопоставлять и выбирать файлы, соответствующие определенным паттернам, и значительно упрощают работу с наборами файлов.

3. Какие операторы управления действиями вы знаете?

В Bash и других оболочках Unix/Linux доступны следующие операторы управления действиями:

- Условные операторы:

    - `if`, `elif`, `else`
- Циклы:
    - `for`
    - `while`
    - `until`
- Операторы прерывания и управления:
    - `break`
    - `continue`
    - `exit`
    - `return`
- Логические операторы:
    - `&&` (логическое И)
    - `||` (логическое ИЛИ)

4. Какие операторы используются для прерывания цикла?

Для прерывания циклов в Bash используются операторы:

- `break` — немедленно завершает выполнение текущего цикла.
- `continue` — пропускает оставшиеся команды в текущей итерации цикла и переходит к следующей итерации.

5. Для чего нужны команды false и true?

Команды `false` и `true` являются встроенными командами в Unix/Linux:

- Команда `true` всегда возвращает статус завершения 0 (успех).
- Команда `false` всегда возвращает статус завершения 1 (ошибка). Эти команды полезны для тестирования, условных операторов и настройки циклов в скриптах.

6. Что означает строка if test -f man$s/$i.$s, встреченная в командном файле?

Строка `if test -f man$s/$i.$s` проверяет, существует ли файл с именем, составленным из значения переменных `$s` и `$i`. Расположение файла — это директория `man$s` и имя файла `$i.$s`. Команда `test -f` проверяет наличие файла и возвращает 0 (успех), если файл существует, и 1 (ошибка), если его нет. Соответствующее действие будет выполнено в зависимости от результата этой проверки.

7. Объясните различия между конструкциями while и until.

Конструкции `while` и `until` в Bash используются для выполнения циклов, но они действуют по-разному:

- `while` цикл: выполняется, пока условие истинно (`true`).

Вывод

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