

Отчет по лабораторной работе

Лабораторная работа 11

Мурзаев Замир Зейнадинович

Содержание

1 Цель работы	5
2 Задание	6
3 Выполнение лабораторной работы	7
4 Выводы	12
Список литературы	13

Список иллюстраций

3.1	Командный файл	7
3.2	файл .c	8
3.3	Командный файл	8
3.4	Командный файл	9
3.5	Командный файл	10

Список таблиц

1 Цель работы

Цель - изучить основы программирования в оболочке ОС UNIX, научиться писать более сложные командные файлы с использованием логических управляемых конструкций и циклов.

2 Задание

1. Используя команды getopt grep, написать командный файл, который анализирует командную строку с ключами: – -iinputfile — прочитать данные из указанного файла; – -ooutputfile — вывести данные в указанный файл; – -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 -р.
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.
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до \square (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).
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).

3 Выполнение лабораторной работы

1)Пишем программу, которая анализирует командную строку, а затем ищет нужные строки (рис. 3.1).

```
#!/bin/bash

#Unpacking parameters
while getopts i:o:p:Cn flag
do
    case $flag in
        i) inputFile=$OPTARG;;
        o) outputFile=$OPTARG;;
        p) pattern=$OPTARG;;
        C) C='--color=always'; echo Flag -$flag will switch color output on;;
        n) n=n;;
        *) echo Illegal option $flag used!;;
    esac
done

touch $outputFile
grep $C -${n}e $pattern $inputFile > $outputFile
```

Рис. 3.1: Командный файл

2)Пишем код на языке С (рис. 3.2) и командный файл, который взаимодействует с ним (рис. 3.3).

The screenshot shows a terminal window with the following details:

- Title bar: prog2.c
- Path: ~/work/study/2022-2023/Операционные системы
- Code content:

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    printf("Введите число: ");
    int a;
    scanf("%d", &a);
    if (a<0) exit(0);
    if (a>0) exit(1);
    if (a==0) exit(2);
    return 0;
}
```

Рис. 3.2: файл .c

The screenshot shows a terminal window with the following details:

- Title bar: prog2.sh - GNU Emacs at dk2n24
- Menu bar: File Edit Options Buffers Tools Sh-Script Outline Hide/Show
- Code content:

```
#!/bin/bash

gcc prog2.c -o prog2
./prog2
code=$?
case $code in
  0) echo "Число меньше 0";;
  1) echo "Число больше 0";;
  2) echo "Число равно 0";;
esac
```

Рис. 3.3: Командный файл

3)Пишем файл, который генерирует определенное количество файлов (рис.

3.4).

The screenshot shows a terminal window with the title bar "prog3.sh". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", "Outline", "Hide/Show", and "Help". The main area contains the following bash script code:

```
#!/bin/bash

opt=$1;
form=$2;
num=$3;
function Files() {
    for ((i=1; i≤$num; i++)) do
        file=$(echo $form | tr '#' "$i")
        if [ $opt = "-r" ]
        then
            rm -f $file
        elif [ $opt = "-c" ]
        then
            touch $file
        fi
    done
}
Files
```

Рис. 3.4: Командный файл

4) Пишем файл, который запаковывает в архив все файлы в директории (рис. 3.5).

prog4.sh - GNU

File Edit Options Buffers Tools Sh-Script

```
#!/bin/bash

files=$(find ./ -maxdepth 1 -mtime -7)
listing=""
for file in "$files" ; do
    file=$(echo "$file" | cut -c 3-)
    listing="$listing $file"
done
dir=$(basename $(pwd))
tar -cvf $dir.tar $listing
```

Рис. 3.5: Командный файл

#Ответы на вопросы

Каково предназначение команды getopt?

Ответ: команда анализирует аргументы, переданные скрипту.

Какое отношение метасимволы имеют к генерации имён файлов?

Ответ: метасимволы позволяют создавать файлы, используя шаблоны.

Какие операторы управления действиями вы знаете?

Ответ: && и ||.

Какие операторы используются для прерывания цикла?

Ответ: break, continue.

Для чего нужны команды false и true?

Ответ: false всегда возвращает код завершения, не равный нулю (т. е. ложь).

Команда true выполняет обратное действие.

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

Ответ: це проверка условия на наличие файла с шаблоном имени if test -f mans/i.\$s.

Объясните различия между конструкциями while и until.

Ответ: при замене в операторе цикла while служебного слова while на until условие, при выполнении которого осуществляется выход из цикла, меняется на противоположное. В остальном оператор цикла while и оператор цикла until идентичны.

4 Выводы

Изучены основы программирования в оболочке ОС UNIX и приобретены навыки по написанию более сложных командных файлов с использованием логических управляющих конструкций и циклов.

Список литературы