

Отчёт лабораторной работы №11

Ветвления и циклы

Баротов Комрон НБИ-01

Содержание

Цель работы	5
Выполнение лабораторной работы	6
Выводы	9

Список таблиц

Список иллюстраций

1	Скрит 1	6
2	Запуск 1	6
3	Скрит 2	7
4	Запуск 2	7
5	Скрит 3	7
6	Запуск 3	8
7	Скрит 4	8
8	Запуск 4	8

Цель работы

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

Выполнение лабораторной работы

1. Скрипт 1 (рис. 1 [-@fig:001])

```
#/bin/bash

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

if [ $m ]
then
    let i=$((i+1))
else
    grep -n -L $mask $input > $output
fi

elif
    grep -l $mask $input > $output
else
    fi
fi
```

Рис. 1: Скрит 1

1. Запуск 1 (рис. 1.1 [-@fig:001])

[illegible]

Рис. 2: Запуск 1

2. Скрипт 2 (рис. 2 [-@fig:002])

```

Открыть  script2
#!/bin/bash
CXX=g++
EXEC=compare
SRC=compare.cpp

if [ "$SRC" -nt "$EXEC" ]
then
    echo "REBUILDING $EXEC..."
    g++ -o $EXEC $SRC
fi

$EXEC $1
ec=$?
if [ "$ec" == "1" ]
then
    echo "Input > 0"
fi
if [ "$ec" == "2" ]
then
    echo "Input == 0"
fi
if [ "$ec" == "3" ]
then
    echo "Input < 0"
fi

```

Рис. 3: Скрипт 2

2. Запуск 2 (рис. 2.2 [-@fig:002])

```

konur@konurab-virtualbox:~$ ./script2 0
REBUILDING compare...
./script2: строка 24: ./EXEC: нет такого файла или каталога
konur@konurab-virtualbox:~$ ./script2 0
./script2: строка 24: ./EXEC: нет такого файла или каталога
konur@konurab-virtualbox:~$ ./script2 0
Input < 0
konur@konurab-virtualbox:~$ ./script2 0
Input == 0
konur@konurab-virtualbox:~$ ./script2 1

```

Рис. 4: Запуск 2

3. Скрипт 3 (рис. 3 [-@fig:003])

```

Открыть  script3
#!/bin/bash
while getopts c:r opt
do
    case $opt in
        c) c="$OPTARG"; for i in $(seq 0 $c); do touch "$i.tmp" >/dev/null;
            r) r="$OPTARG"; name=$(cat /dev/urandom | tr -dc 'a-z' | fold -w $r | xargs echo | sha256sum | cut -d ' ' -f 1); done;
    done
done

```

Рис. 5: Скрипт 3

3. Запуск 3 (рис. 3.3 [-@fig:003])

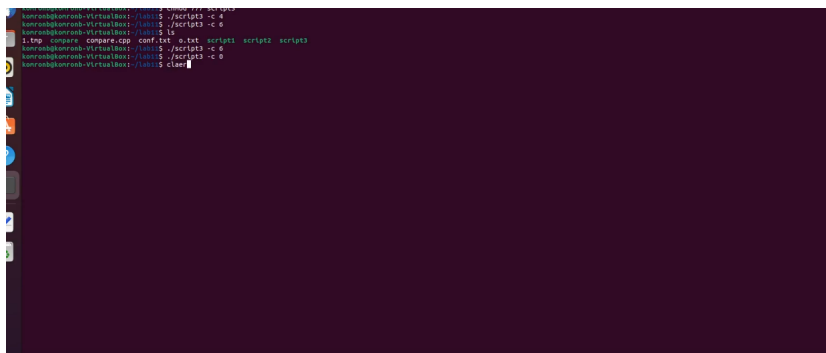


Рис. 6: Запуск 3

4. Скрипт 4 (рис. 4 [-@fig:004])

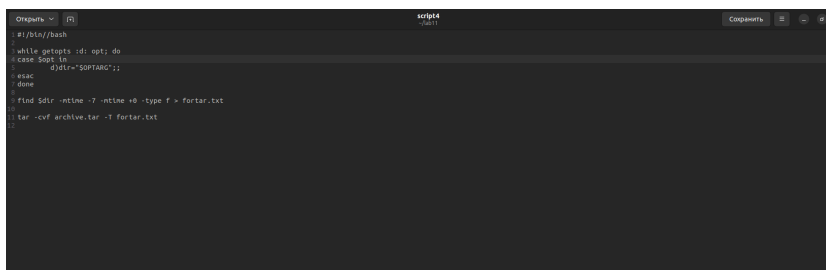


Рис. 7: Скрит 4

4. Запуск 3 (рис. 4.4 [-@fig:004])

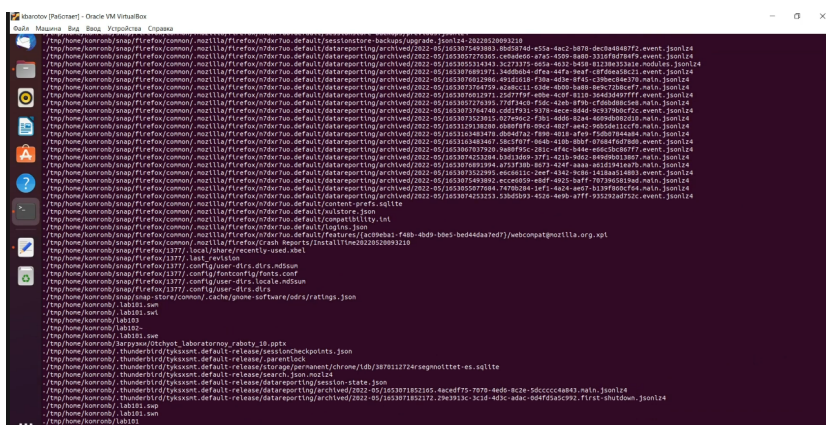


Рис. 8: Запуск 4

Выводы

В данной работе мы научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.