

Лабораторная работа №11

Баротов Комрон НБИ-01

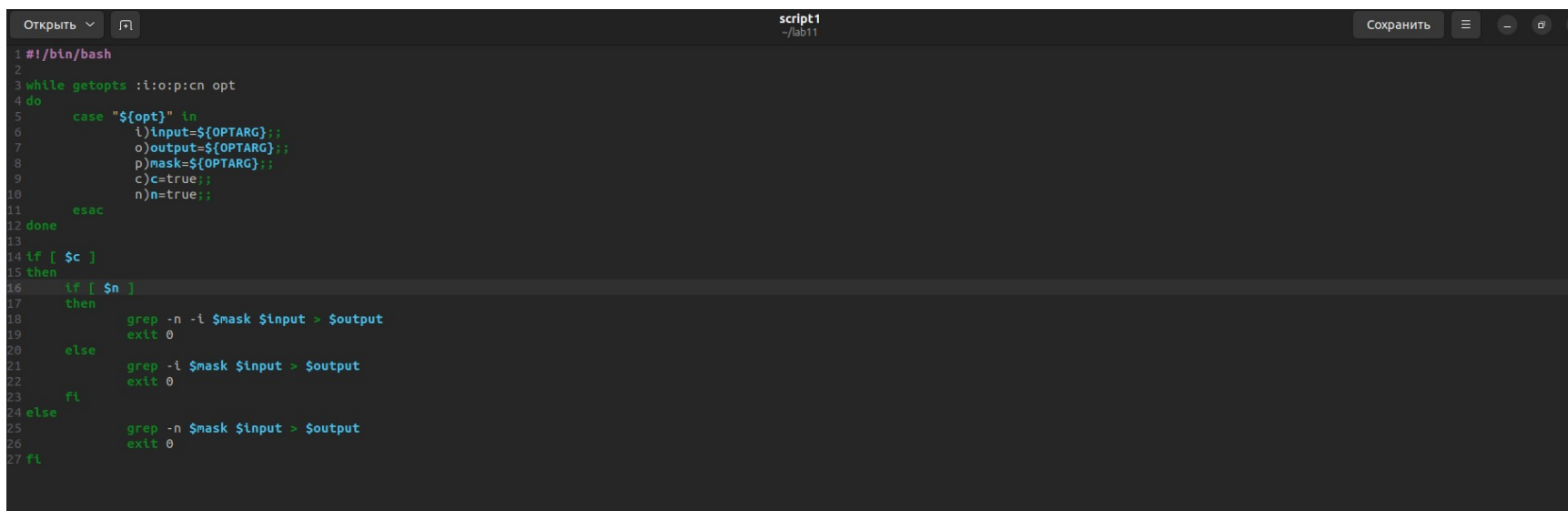
ПРОГРАММИРОВАНИЕ В КОМАНДНОМ ПРОЦЕССОРЕ ОС UNIX/LINUX .ВЕТВЛЕНИЯ И ЦИКЛЫ

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)



The image shows a terminal window with a dark background. The title bar at the top indicates the file is 'script1' located at '~/lab11'. On the left, there are buttons for 'Открыть' (Open) and a file icon. On the right, there are buttons for 'Сохранить' (Save), a menu icon, and window control buttons (minimize, maximize, close). The terminal content is a shell script with line numbers 1 through 27. The script starts with a shebang, then enters a loop to process command-line options. It uses a case statement to handle options: 'i' for input, 'o' for output, 'p' for mask, 'c' for case-insensitive search, and 'n' for normal search. After processing options, it checks if the case-insensitive flag 'c' is set. If so, it enters another conditional block where it checks if the normal flag 'n' is also set. Depending on these flags, it runs 'grep' with appropriate options and exits with status 0. If 'c' is not set, it runs 'grep -n' with the mask. The script ends with 'fi'.

```
1#!/bin/bash
2
3while getopts :i:o:p:cn opt
4do
5    case "${opt}" in
6        i)input=${OPTARG};;
7        o)output=${OPTARG};;
8        p)mask=${OPTARG};;
9        c)c=true;;
10       n)n=true;;
11    esac
12done
13
14if [ $c ]
15then
16    if [ $n ]
17    then
18        grep -n -i $mask $input > $output
19        exit 0
20    else
21        grep -i $mask $input > $output
22        exit 0
23    fi
24else
25    grep -n $mask $input > $output
26    exit 0
27fi
```

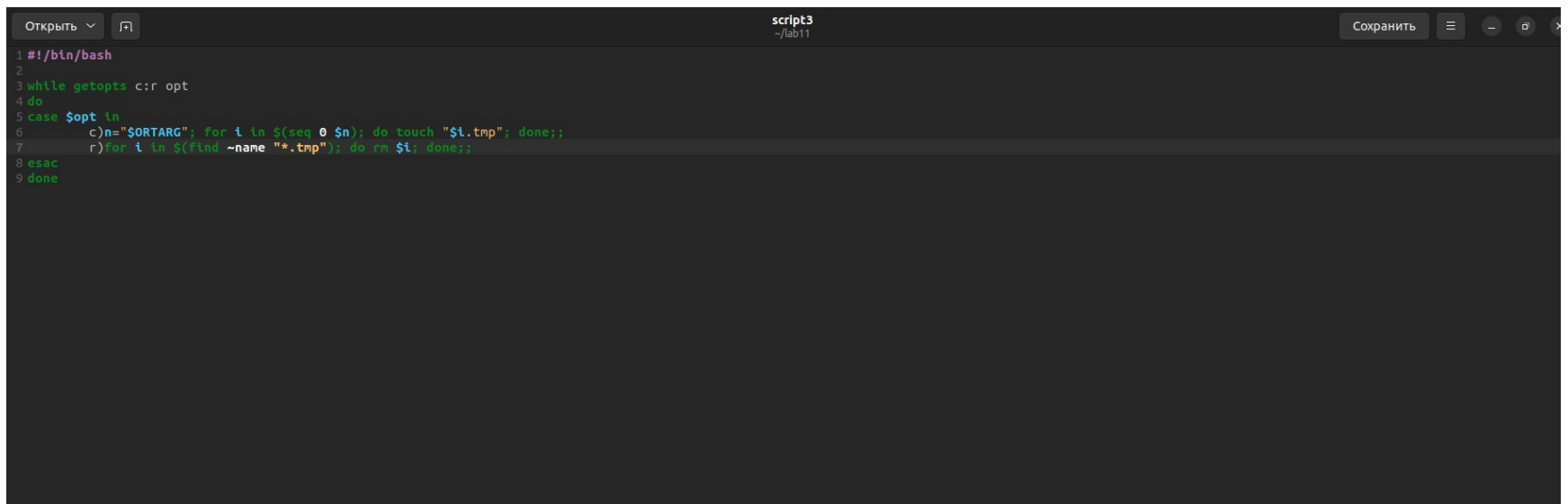
Скрипт 1

2.Скрипт 2 (рис.2)

```
1 #!/bin/bash
2
3 CC=g++
4 EXEC=compare
5 SRC=compare.cpp
6
7 if [ "$SRC" -nt "$EXEC" ]
8
9 then
10     echo "REBUILDING $EXEC..."
11     $CC -o $EXEC $SRC
12 fi
13
14 ./EXEC $1
15
16 ec=$?
17 if [ "$ec" == "1" ]
18 then
19     echo "input > 0"
20 fi
21
22 if [ "$ec" == "2" ]
23 then
24     echo "input == 0"
25 fi
26
27 if [ "$ec" == "3" ]
28 then
29     echo "input < 0"
30 fi
```

Скрипт 2

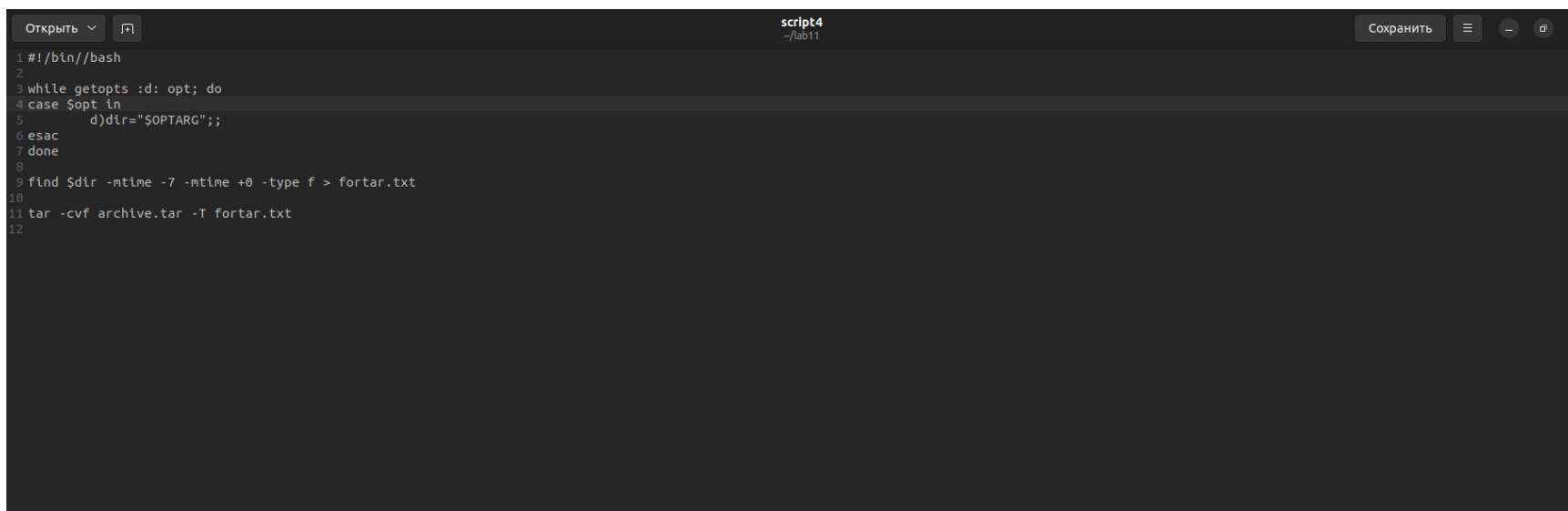
3.Скрипт 3 (рис.3)



```
1#!/bin/bash
2
3while getopts c:r opt
4do
5case $opt in
6    c)n=$OPTARG; for i in $(seq 0 $n); do touch "$i.tmp"; done;;
7    r)for i in $(find ~name "*.tmp"); do rm $i; done;;
8esac
9done
```

Скрипт 3

4.Скрипт 4 (рис.4)



```
1#!/bin/bash
2
3while getopts :d: opt; do
4case $opt in
5    d)dir="$OPTARG";;
6esac
7done
8
9find $dir -mtime -7 -mtime +0 -type f > fortar.txt
10
11tar -cvf archive.tar -T fortar.txt
12
```

Скрипт 4

Вывод

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

Wer's nicht glaubt, bezahlt einen Taler