

Отчёт по лабораторной работе №11

Степанов Иван Юрьевич

апрель 2022

РУДН, Москва, Россия

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

(рис. [-@fig:001])

```
#!/bin/bash
iflag=0; oflag=0 pflag=0; Cflag=0; nflag=0;
do case $optletter in
    i) iflag=1; ival=$OPTARG;;
    o) oflag=1; oval=$OPTARG;;
    p) pflag=1; pval=$OPTARG;;
    C) cflag=1;;
    n) nflag=1;;
    *) echo illegal option $optletter
esac
done
if (($iflag==0))
then echo "файл не найден"
else
    if (($iflag==0))
    then echo "файл не найден"
    else
        if (($oflag==0))
        then if (($Cflag==0))
            then if (($nflag==0))
                then grep $pval $ival
                else grep -n $pval $ival
                fi
            else if (($nflag==0))
                then grep -i $pval $ival
                else grep -i -n $pval $ival
                fi
            fi
        else if (($Cflag==0))
            then if (($nflag==0))
                then grep $pval $ival > $oval
```

(рис. [-@fig:003])

A screenshot of a terminal window. The window has a title bar with several icons: a document with a plus sign, a folder, a floppy disk, a close button (X), a 'Save' button (floppy disk icon), and a yellow arrow pointing left. The terminal content shows a shell prompt and a script. The script compiles a C program, runs it, and then uses a case statement to compare the program's output with zero. The 'Save' button in the title bar is highlighted with a mouse cursor, and a tooltip 'Save current file' is visible.

```
#!/bin/bash

gcc prog2.c -o prog2
./prog2
code=$?
case $code in
    0) echo "chislo menshe 0";;
    1) echo "chislo bolshe 0";;
    2) echo "chislo ravno 0";;
esac
```

Рис. 3: второй скрипт

(рис. [-@fig:005])

```
#!/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
}
```

Рис. 5: третий скрипт

(рис. [-@fig:006])

```
#!/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. 6: четвертый скрипт

я научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Спасибо за внимание!