

Лабораторная работа №11

Степанов Иван Юрьевич

Содержание

Цель работы	5
Выполнение лабораторной работы	6
Выводы	11

Список иллюстраций

0.1	первый скрипт	6
0.2	первый скрипт	7
0.3	второй скрипт	8
0.4	второй скрипт	9
0.5	третий скрипт	10
0.6	четвертый скрипт	10

Список таблиц

Цель работы

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

Выполнение лабораторной работы

первый командный файл

(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
                else
```

Рис. 0.1: первый скрипт

(рис. [-@fig:002])

```

if(($soflag==0))
then if (($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 $pbal $ival
fi
fi
else if (($Cflag==0))
then if (($nflag==0))
then grap $pval $ival > $oval
else grep -n $pval $ival > $oval
fi
else if (($nflag==0))
then grep -i $pval $ival > $oval
else grep -i -n $pval $ival > $oval
fi
fi
fi
fi

```

Рис. 0.2: первый скрипт

второй командный файл

(рис. [-@fig:003])



Рис. 0.3: второй скрипт

(рис. [-@fig:004])


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

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

Рис. 0.4: второй скрипт

третий командный файл

(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. 0.5: третий скрипт

четвертый командный файл

(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. 0.6: четвертый скрипт

Выводы

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