

Отчёт по лабораторной работе №11

Операционные системы

Пинега Белла Александровна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	11
	Список литературы	12

Список иллюстраций

Список таблиц

1 Цель работы

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

2 Задание

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами: `-iinputfile` — прочитать данные из указанного файла; `-ooutputfile` — вывести данные в указанный файл; `-r` — 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 `-r`.
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 N (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 `tag` 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 Теоретическое введение

4 Выполнение лабораторной работы

1. Используя команды `getopts` `grep`, я написала командный файл, который ана-

```
#!/bin/bash
iflag=0; oflag=0; pflag=0; cflag=0; nflag=0;
while getopts i:o:p:C:n optletter
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 $optletter
    esac
done
if (($pflag==0))
then echo "Шаблон 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
        fi
    fi
fi
```

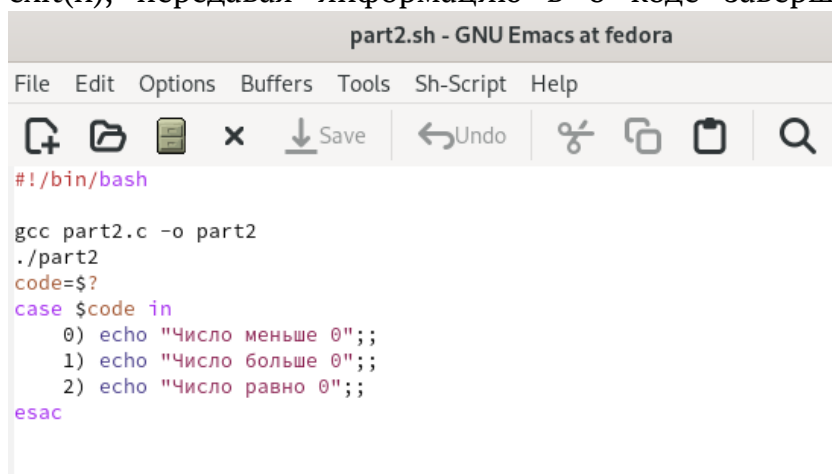
лизирует командную строку с ключами:


```

        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 grep $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 $ibal > $oval
fi
fi
fi
fi
fi

```

2. Затем я написала 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:



```

part2.sh - GNU Emacs at fedora
File Edit Options Buffers Tools Sh-Script Help
+ Save Undo % 
#!/bin/bash

gcc part2.c -o part2
./part2
code=$?
case $code in
    0) echo "Число меньше 0";;
    1) echo "Число больше 0";;
    2) echo "Число равно 0";;
esac

```

3. Далее я написала командный файл, создающий указанное число файлов,

```
part3.sh - GNU Emacs at  
File Edit Options Buffers Tools Sh-Script Help  
[Icons] Save Undo [Icons]  
#!/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
```

пронумерованных последовательно от 1 до N :

4. И последним шагом я написала командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории:

```
part4.sh - GNU Emacs at fedora  
File Edit Options Buffers Tools Sh-Script Help  
[Icons] Save Undo [Icons]  
#!/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
```

5 Выводы

Я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Список литературы