

Лабораторная работа 12

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Кузнецов Юрий Владимирович

Содержание

Цель работы	1
Задание	1
Выполнение лабораторной работы.....	1
Выводы.....	3

Цель работы

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

Задание

- написать 4 командных файла.
- написать программу на Си.
- работа с файлом .txt.

Выполнение лабораторной работы

Ознакомился с материалами лабораторной работ и приступил к выполнению заданий.

Используя команды `getopts` `grep`, написал командный файл (рис. 1), который анализирует командную строку с ключами `-i`, `-o`, `-p`, `-C`, `-n`. Сначала создал текстовый файл (рис. 2). (рис. 3)

```
#!/bin/bash
while getopts i:o:p:Cn 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 Illegaloption $optletter
    esac
done
if(((Cflag==1)&&(nflag==1)))
then grep -e${pval} -i -n ${ival}
    if ((oflag==1))
    then grep -e${pval} -i -n ${ival} > ${oval}
    fi
fi
if(((Cflag==1)&&(nflag==0)))
then grep -e${pval} -i ${ival}
    if ((oflag==1))
    then grep -e${pval} -i ${ival} > ${oval}
    fi
fi
if(((Cflag==0)&&(nflag==1)))
then grep -e${pval} -n ${ival}
    if ((oflag==1))
    then grep -e${pval} -n ${ival} > ${oval}
    fi
fi
if(((Cflag==0)&&(nflag==0)))
then grep -e${pval} ${ival}
    if ((oflag==1))
    then grep -e${pval} ${ival} > ${oval}
    fi
fi
fi
```

Командный файл

```
hello how
are you
yellow submarine
good luck!
good morning.
```

Текстовый файл

Написал на языке Си программу (рис. 5), 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 о коде завершения в оболочку. Командный файл (рис. 4) вызывает эту программу и, проанализировав с помощью команды `$?`, выдал сообщение о том, какое число было введено. (рис. 6)

```
#!/bin/bash
gcc -o cprog failci.c
./cprog
case $? in
    0)echo 'Это число равно 0';;
    1)echo 'Это число меньше нуля';;
    2)echo 'Это число больше 0';;
esac
```

Командный файл

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int x;
6     printf("Введите число: ");
7     scanf("%i", &x);
8     if (x==0){
9         exit(0);}
10    else if (x<0){
11        exit(1);}
12    else if (x>0){
13        exit(2);}
14    return (3);
15 }

```

Написал командный файл (рис. 7), создающий указанное число файлов, пронумерованных последовательно от 1 до N. Число файлов, которые было необходимо создать, передавалось в аргументы командной строки. Этот же командный файл умеет удалять все созданные им файлы (если они существуют). (рис. 8)

```

#!/bin/bash
let dflag=0;
while getopts t:r optletter
do case $optletter in
    t) aflag=1; aval=$OPTARG;;
    r) dflag=1;;
    *) echo illegaloption $optletter
    esac
done
#echo ${aval}
if((dflag==0))
then for ((i=1;i<=aval;i++))
do touch ${i}.tmp
done
fi
if ((dflag==1))
then for ((i=1;i<=aval;i++))
do rm ${i}.tmp
done
fi

```

Командный файл

Написал командный файл (рис. 9), который с помощью команды tar запаковывает в архив все файлы в указанной директории.

```

#!/bin/bash
tar -cf 12.tar $@
tar -cf tarr.tar
find $@ -mtime -14 -exec tar -rf tarr.tar '{}' ';'

```

Командный файл

Выводы

- Я успешно изучил основы программирования в оболочке ОС UNIX.
- Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.