

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ № 12.

дисциплина:      *Операционные системы*

Студент: Абушек Дмитрий Олегович

Группа: НФИбд-01-20

МОСКВА

2021 г.

**Цель работы:** изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

### Ход работы.

1. Используя команды `getopts` `grep` (предварительно создала текстовый файл `file.txt` и скрипт `lab12.sh`), напишем командный файл (рис.1-2), который анализирует командную строку с ключами:

- `-iinputfile` — прочитать данные из указанного файла;
- `-ooutputfile` — вывести данные в указанный файл;
- `-ршаблон` — указать шаблон для поиска;
- `-C` — различать большие и малые буквы;
- `-n` — выдавать номера строк.

```
File Edit Options Buffers Tools Sh-Script Help
do case $optletter in
    o) oflag=1; oval=$OPTARG;;
    i) iflag=1; ival=$OPTARG;;
    L) lflag=1;;
    t) tflag=1;;
    r) rflag=1;;
    *) echo Illegal option $optletter
esac
done
if (((Cflag==1)&&(nflag==0)))
then grep -e${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} > ${oval}
    fi
fi
if (((Cflag==1)&&(nflag==1)))
then grep -e${pval} -i -n ${ival}
    if ((oflag==1))
    then grep -e${pval} -i -n ${ival} > ${oval}
    fi
fi
if (((Cflag==0)&&(nflag==0)))
then grep -e${pval} ${ival}
    if ((oflag==1))
    then grep -e${pval}
        ${ival} > ${oval}
    fi
fi
fi
```

2. Написали 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 коде завершения в оболочку (рис.3-5);

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

int main() {
    int x;
    printf("input: ");
    scanf("%i", &x);
    if (x==0) exit (0);
    else if (x<0) exit (1);
    else if (x>0) exit (2);
    return(3);
}
```

```
#!/bin/bash
gcc -o cprog lab12.c
./cprog
case $? in
    0) echo 'equal zero';;
    1) echo 'negativ';;
    2) echo 'positiv';;
esac
```

```
dmitryabushek@mbp-dmitrij ~ % bash lab12c.sh
input: 0
equal zero
dmitryabushek@mbp-dmitrij ~ % bash lab12c.sh
input: 1
positiv
dmitryabushek@mbp-dmitrij ~ % bash lab12c.sh
input: -4
negativ
```

3. Напишем 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) (рис.6-8);

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

[dmityabushek@mbp-dmitrij ~ % sh lab123.sh -a4
[dmityabushek@mbp-dmitrij ~ % ls -l
total 272
-rw-r--r--    1 dmityabushek  staff    97 12 сен 00:10 #lab07.sh#
-rw-r--r--    1 dmityabushek  staff     0 12 сен 21:17 0.txt
drwx-----@   3 dmityabushek  staff    96 24 сен  2020 Applications
drwx-----@  25 dmityabushek  staff   800 12 сен 18:58 Desktop
drwx-----+   6 dmityabushek  staff   192  9 сен 15:00 Documents
drwx-----@ 197 dmityabushek  staff  6304 12 сен 18:21 Downloads
drwx-----@  74 dmityabushek  staff  2368 19 июн 14:44 Library
drwx-----+  16 dmityabushek  staff   512 12 сен 19:28 Movies
drwx-----+   4 dmityabushek  staff   128  8 сен  2020 Music
drwx-----@  33 dmityabushek  staff  1056 11 сен 20:28 OneDrive - Российский У
ниверситет Дружбы Народов
drwx-----+   6 dmityabushek  staff   192 25 ноя  2020 Pictures
drwxr-xr-x+   4 dmityabushek  staff   128  6 сен  2020 Public
drwxr-xr-x  14 dmityabushek  staff   448 28 дек  2020 Qt
drwxr-xr-x    3 dmityabushek  staff    96 29 апр 19:08 VirtualBox VMs
```

4. Напишем командный файл, который с помощью команды tar 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) (рис.9-10);

```
#!/bin/bash
tar -cf 12.tar $@
tar -cf 12l.tar
find $@ -mtime -7 -exec tar -rf 12l.tar '{}' ';'

[dmityabushek@mbp-dmitrij ~ % bash lab124.sh /monthly
tar: /monthly: Cannot stat: No such file or directory
tar: Error exit delayed from previous errors.
tar: no files or directories specified
find: /monthly: No such file or directory
[dmityabushek@mbp-dmitrij ~ % ls
#lab07.sh#
0.txt
12.tar
```

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

### **Ответы на контрольные вопросы:**

1. Команда `getopts` является встроенной командой командной оболочки `bash`, предназначенной для разбора параметров сценариев. Она обрабатывает исключительно однобуквенные параметры как с аргументами, так и без них и этого вполне достаточно для передачи сценариям любых входных данных.

2. При генерации имен используют метасимволы:

- \* произвольная (возможно пустая) последовательность символов;

- ? один произвольный символ;

- [...] любой из символов, указанных в скобках перечислением и/или с указанием диапазона;

- `cat f*` выдаст все файлы каталога, начинающиеся с "f";

- `cat *f*` выдаст все файлы, содержащие "f";

- `cat program.?` выдаст файлы данного каталога с однобуквенными расширениями, скажем "program.c" и "program.o", но не выдаст "program.com";

- `cat [a-d]*` выдаст файлы, которые начинаются с "a", "b", "c", "d". Аналогичный эффект дадут и команды `cat [abcd]*` и `cat [bdac]*`.

3. Операторы `&&` и `||` являются управляющими операторами. Если в командной строке стоит `command1 && command2`, то `command2` выполняется в том, и только в том случае, если статус выхода из команды `command1` равен нулю, что говорит об успешном ее завершении. Аналогично, если командная строка имеет вид `command1 || command2`, то команда `command2` выполняется тогда, и только тогда, когда статус выхода из команды `command1` отличен от нуля.

4. Оператор `break` завершает выполнение ближайшего включающего цикла или условного оператора, в котором он отображается.

5. Команда `true` всегда возвращает ноль в качестве выходного статуса для индикации успеха. Команда `false` всегда возвращает не-ноль в качестве выходного статуса для индикации неудачи. Во всех управляющих конструкциях в качестве логического значения используется код возврата из программы, указанной в качестве условия. Код возврата 0 – истина, любое

другое значение – ложь. Программа true – всегда завершается с кодом 0, false – всегда завершается с кодом 1.

6. Введенная строка означает условие существования файла man\$s/\$i.\$s

7. Цикл While выполняется до тех пор, пока указанное в нем условие истинно. Когда указанное условие становится ложным - цикл завершается. Цикл Until выполняется до тех пор, пока указанное в нем условие ложно.