

Лабораторная работа 12

Программирование в командном процессоре ОС UNIX. Расширенное
программирование

Гарут Александр Игоревич

Содержание

Цель работы	3
Ход работы	4
Вывод	7
Контрольные вопросы	8

Цель работы

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

Ход работы

1. Выполним пункт 1 лабораторной работы

```
lockfile="./locking.file"

exec {fn}>"$lockfile"
if test -f "$lockfile"
then
    while [ 1!=0 ]
    do
        if flock -n ${fn}
        then echo "file was locked"
            sleep 4
            echo "unlocking"
            flock -u ${fn}

        else
            echo "file already locked"
```

Изображение1.1: Код скрипта

Удостовермся что всё работает

```
[garutalek@fedora lab12]$ ./lab12 1
file was locked
unlocking
file was locked
unlocking
file was locked
unlocking
file was locked
```

Изображение1.2: Запуск скрипта

2. Напишем скрипт, аналог команды man

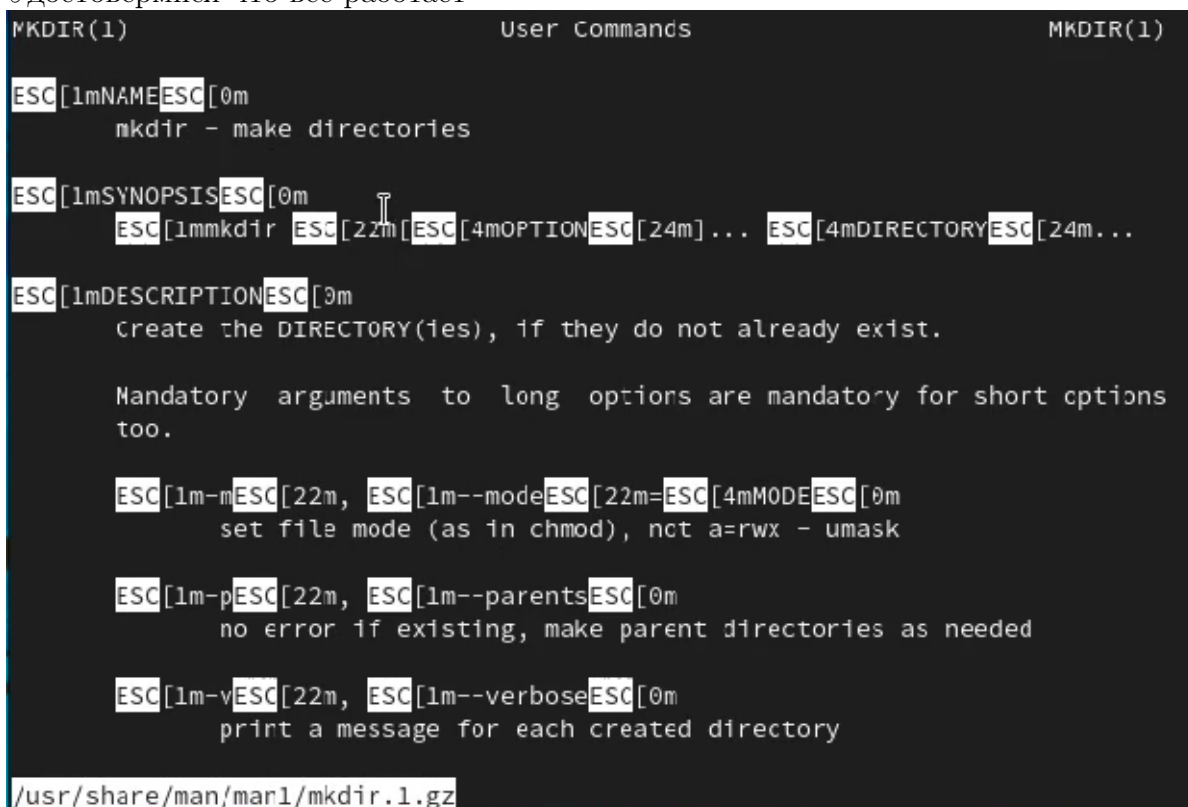
```
command=""

while getopts :n: opt
do
case $opt in
n)command="$OPTARG";;
esac
done

if test -f "/usr/share/man/man1/$command.1.gz"
then less /usr/share/man/man1/$command.1.gz
else
echo "not found command"
fi
```

Изображение2.1: Код скрипта

Удостовермся что всё работает



```

MKDIR(1)                                User Commands                                MKDIR(1)

ESC[1mNAMEESC[0m
    mkdir - make directories

ESC[1mSYNOPSISESC[0m
    ESC[1mmkdir ESC[22m[ESC[4mOPTIONESC[24m]... ESC[4mDIRECTORYESC[24m...

ESC[1mDESCRIPTIONESC[0m
    Create the DIRECTORY(ies), if they do not already exist.

    Mandatory arguments to long options are mandatory for short options
    too.

    ESC[1m-mESC[22m, ESC[1m--modeESC[22m=ESC[4mMODEESC[0m
        set file mode (as in chmod), nct a=rwx - umask

    ESC[1m-pESC[22m, ESC[1m--parentsESC[0m
        no error if existing, make parent directories as needed

    ESC[1m-vESC[22m, ESC[1m--verboseESC[0m
        print a message for each created directory

/usr/share/man/man1/mkdir.1.gz
```

Изображение2.2: Запуск скрипта

3. Напишем скрипт, выдающий случайный набор символов

```
echo $RANDOM | tr '0-9' 'a-zA-Z'
```

Изображение3.1: Код скрипта

Удостовермся что всё работает

```
[garutalek@fedora lab12]$ ./lab12_3  
dgic  
[garutalek@fedora lab12]$ ./lab12_3  
degf
```

Изображение3.2: Запуск скрипта

Вывод

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

Контрольные вопросы

1. Нужно поместить выражение в $(())$
2. $\$str1 + \$str2$