

Лабораторная работа №12. Программирование в командном процессоре ОС UNIX. Расширенное программирование

Латыпова Диана. НФИбд-02-21

¹RUDN University, Moscow, Russian Federation

Цель работы

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

Выполнение лабораторной работы

1 задание

Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени t_1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени $t_2 < t_1$, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустить командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой ($> /dev/tty\#$, где $\#$ — номер терминала куда перенаправляется вывод), в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов.

```
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 2
    echo "Unlocking"
    flock -u ${fn}
else
    echo "File already locked"
    sleep 2
fi
done

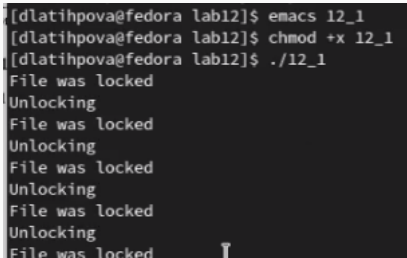
fi
```

1 задание

Сначала предоставила права на выполнение файла 12_1. И запустила файл(рис. 1):

```
1 chmod +x 12_1
```

```
2 ./12_1
```



```
[dlatihpova@fedora lab12]$ emacs 12_1
[dlatihpova@fedora lab12]$ chmod +x 12_1
[dlatihpova@fedora lab12]$ ./12_1
File was locked
Unlocking
File was locked
Unlocking
File was locked
Unlocking
File was locked
Unlocking
File was locked
```

Figure 1: Результат 1 скрипта

Реализовать команду `man` с помощью командного файла. Изучите содержимое каталога `/usr/share/man/man1`. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой `less` сразу же просмотрев содержимое справки. Командный файл должен получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге `man1`.

```
command=""

while getopts :m: opt
do

    case $opt in
        m)command="$OPTARG";;
        esac

done

if test -f "/usr/share/man/man1/$command.1.gz"
then less /usr/share/man/man1/$command.1.gz
fi
```

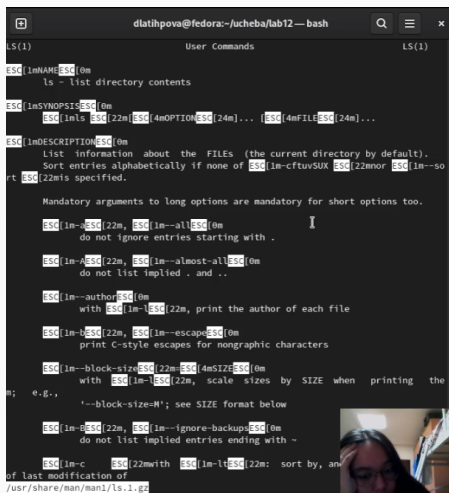
2 задание

Снова предоставила права на выполнение файла 12_2. И запустила файл(рис. 2)(рис. 3):

```
[dlatihpova@fedora lab12]$ emacs 12_2  
[dlatihpova@fedora lab12]$ chmod +x 12_2  
[dlatihpova@fedora lab12]$ ./12_2 -m ls
```

Figure 2: Результат 2 скрипта(1)

2 задание



```
LS(1)                                User Commands                                LS(1)

ESC[1mNAMEESC[0m
    ls - list directory contents

ESC[1mSYNOPSISESC[0m
    ESC[1m[ls ESC[22m[ESC[4mOPTIONESC[24m]... [ESC[4mFILEESC[24m]...

ESC[1mDESCRIPTIONESC[0m
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of ESC[1m-cftuvSUX ESC[22mnor ESC[1m--so
rt ESC[22mis specified.

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

    ESC[1m-aESC[22m, ESC[1m--allESC[0m
        do not ignore entries starting with .

    ESC[1m-AESC[22m, ESC[1m--almost-allESC[0m
        do not list implied . and ..

    ESC[1m--authorESC[0m
        with ESC[1m-lESC[22m, print the author of each file

    ESC[1m-bESC[22m, ESC[1m--escapeESC[0m
        print C-style escapes for nongraphic characters

    ESC[1m--block-sizeESC[22m[ESC[4mSIZEESC[0m
        with ESC[1m-ESC[22m, scale sizes by SIZE when printing the
m; e.g.,
        '--block-size=M'; see SIZE format below

    ESC[1m-BESC[22m, ESC[1m--ignore-backupsESC[0m
        do not list implied entries ending with ~

    ESC[1m-c ESC[22mwith ESC[1m-ltESC[22m: sort by, and
of last modification of
/usr/share/man/man1/ls.1.gz
```

Figure 3: Результат 2 скрипта(2)

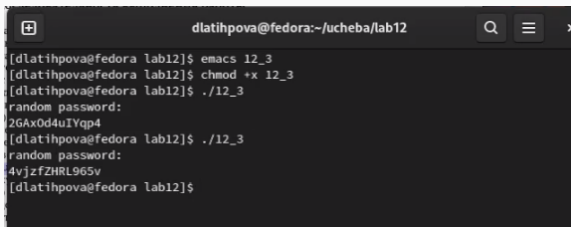
Используя встроенную переменную `$RANDOM`, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что `$RANDOM` выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

```
echo "random password:"
```

```
cat /dev/urandom | tr -dc "a-zA-Z0-9" | fold -w 13 | head -n 1
```

3 задание

Предоставила права на выполнение файла 12_3. И запустила файл(рис. 4):

A terminal window with a dark background. The title bar shows the user 'dlatihpova' at 'fedora' in the directory '~/учеба/lab12'. The terminal contains the following text:

```
[dlatihpova@fedora lab12]$ emacs 12_3
[dlatihpova@fedora lab12]$ chmod +x 12_3
[dlatihpova@fedora lab12]$ ./12_3
random password:
2GAx0d4uIYqp4
[dlatihpova@fedora lab12]$ ./12_3
random password:
4vjzfZHRL965v
[dlatihpova@fedora lab12]$
```

Figure 4: Результат 3 скрипта

Выводы

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