

Отчёт по лабораторной работе №10

Программирование в командном процессоре ОС UNIX.

Расширенное программирование

Архипов Юрий Денисович

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. 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 выход в другой ($> /dev/tty\#$, где $\#$ - номер терминала, куда перенаправляется выход), в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов.

а) Создаём файл для скрипта и файл, с которым будет работать скрипт и запускаем emacs, после меняем права доступа.

Рис.1

```
[yurallarhipov@yurallarhipov ~]$ touch l.sh
[yurallarhipov@yurallarhipov ~]$ touch l.file
[yurallarhipov@yurallarhipov ~]$ emacs l.sh
[yurallarhipov@yurallarhipov ~]$ chmod + l.sh
```

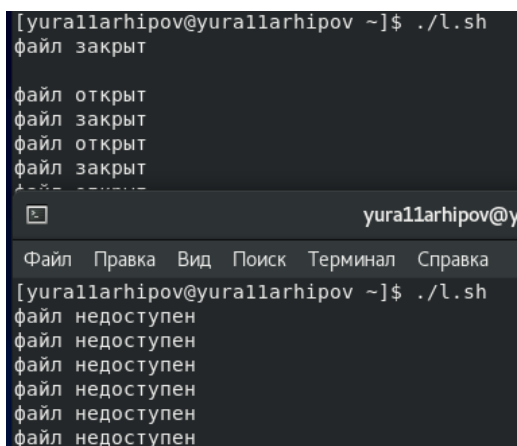
б) Пишем скрипт в первый файл.

```
#!/bin/bash

lockfile="./l.file"
exec {fn}>$lockfile
if test -f "$lockfile"
then
    while [ 1 = 1 ]
    do
        if flock -n ${fn}
        then
            echo "файл закрыт"
            sleep 5
            echo "файл открыт"
            flock -u ${fn}
        else
            echo "файл недоступен"
            sleep 5
        fi
    done
else
    echo "файла нет"
fi
```

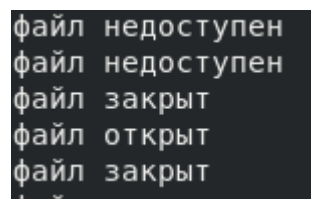
Рис.2

с) Запускаем файл в первом терминале, затем во втором и проверяем исправность скрипта.



```
[yurallarhipov@yurallarhipov ~]$ ./l.sh
файл закрыт
файл открыт
файл закрыт
файл открыт
файл закрыт
файл недоступен
файл недоступен
файл недоступен
файл недоступен
файл недоступен
файл недоступен
```

Рис.3



```
файл недоступен
файл недоступен
файл закрыт
файл открыт
файл закрыт
```

рис.4

2. Реализовать команду man с помощью командного файла. Изучить 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.

а) Создаём файл для скрипта и открываем его в emacs, после меняем права пользователя.

Рис.5

```
[yurallarhipov@yurallarhipov ~]$ touch man.sh
[yurallarhipov@yurallarhipov ~]$ emacs man.sh
[yurallarhipov@yurallarhipov ~]$ chmod +x man.sh
```

б) Пишем скрипт.

Рис.6

```
#!/bin/bash

name=""

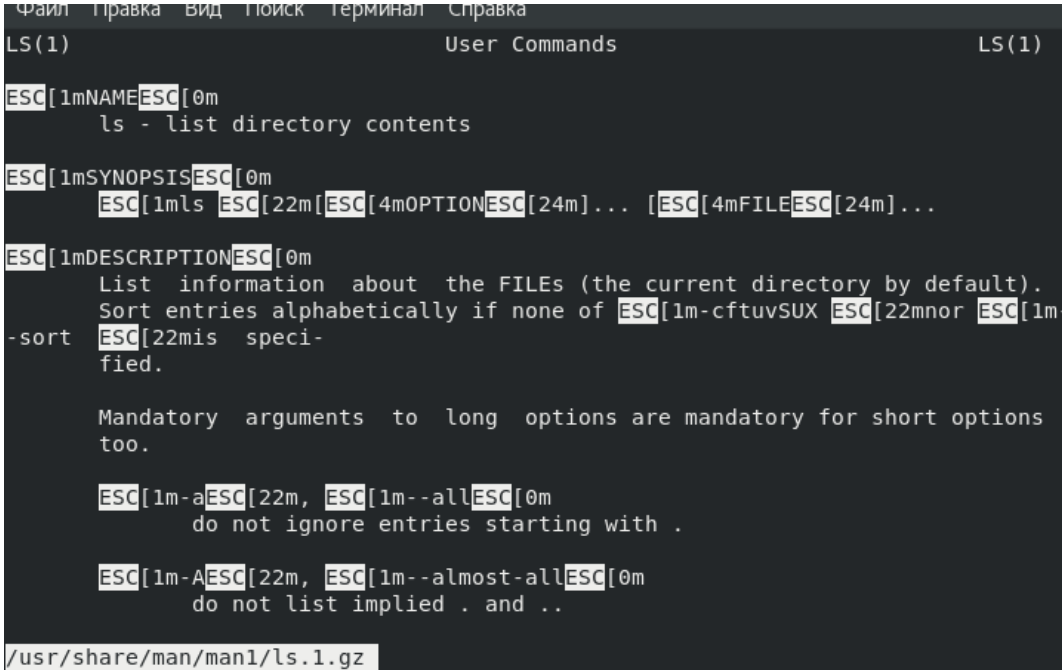
while getopts "n:" opt
do
    case $opt in
        n) name="$OPTARG";;
    esac
done
if test -f "/usr/share/man/man1/$name.1.gz"
then
    less /usr/share/man/man1/"$name".1.gz
else
    echo "Нет справки об этой команде"
fi
```

с) Запускаем файл и проверяем исправность скрипта.

Рис.7

```
[yurallarhipov@yurallarhipov ~]$ ./man.sh -n ls
```

Рис.8



```
Файл Правка Вид Поиск Терминал Справка
LS(1) User Commands LS(1)
ESC[1mNAMEESC[0m
ls - list directory contents
ESC[1mSYNOPSISESC[0m
ESC[1mls 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-
-sort ESC[22mis speci-
fied.
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 ..
/usr/share/man/man1/ls.1.gz
```

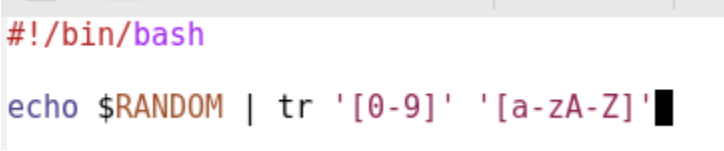
3. используя встроенную переменную \$RANDOM, написать командный файл, генерирующий случайную последовательность букв латинского алфавита.

а) Создаём файл для скрипта и открываем его в emacs, после меняем права пользователя.

Рис.9 

```
[yurallarhipov@yurallarhipov ~]$ touch random.sh
[yurallarhipov@yurallarhipov ~]$ emacs random.sh
[yurallarhipov@yurallarhipov ~]$ chmod +x random.sh
```

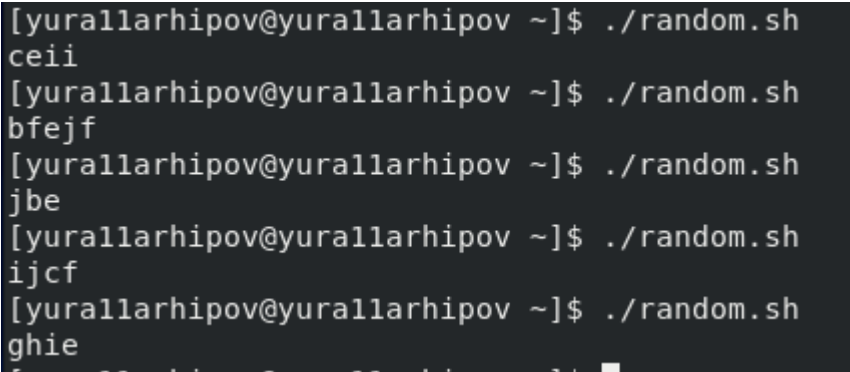
б) Пишем скрипт.

Рис.10 

```
#!/bin/bash

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

с) Запускаем файл и проверяем исправность скрипта.

Рис.11 

```
[yurallarhipov@yurallarhipov ~]$ ./random.sh
ceii
[yurallarhipov@yurallarhipov ~]$ ./random.sh
bfejf
[yurallarhipov@yurallarhipov ~]$ ./random.sh
jbe
[yurallarhipov@yurallarhipov ~]$ ./random.sh
ijcf
[yurallarhipov@yurallarhipov ~]$ ./random.sh
ghie
```

Вывод

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