

Презентация по лабораторной работе №13

Denis A. Seregin^{1,3}

2021, 05 June

¹RUDN University, Moscow, Russian Federation

Содержание

- Цель работы
- Задачи
- Результаты
- Вывод

Цель работы

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

Основные понятия

- Написать командный файл, реализующий упрощённый механизм семафоров.
- Реализовать команду `map` с помощью командного файла.
- Используя встроенную переменную `$RANDOM`, написать командный файл, генерирующий случайную последовательность букв латинского алфавита.

Результаты

Первая задача

Написал 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.

```
#!/bin/bash
function func1
{
touch lr13.txt
ls>lr13.txt
for((i=0;i<5;i++))
do
echo "скрипт работает"
sleep 2
done
rm lr13.txt
}
function func2
{
while test -f lr13.txt
do
echo "занято другим процессом"
sleep 2
done
}
func2
func1
```

Результаты работы

```
[root@localhost daseregin]# tty
/dev/pts/1
[root@localhost daseregin]# ./lab13.sh
скрипт работает
занято другим процессом
скрипт работает
занято другим процессом
скрипт работает
занято другим процессом
скрипт работает
занято другим процессом
скрипт работает
занято другим процессом
[root@localhost daseregin]# скрипт работает
скрипт работает
скрипт работает
```

Figure 2: Вывод в первом терминале

```
[root@localhost daseregin]# ./lab13.sh>/dev/pts/1 &
[1] 5416
[root@localhost daseregin]# █
```

Figure 3: Вывод во втором терминале

Вторая задача

Реализовал 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`.

```
daseregin@localhost:/home/daseregin
Файл  Правка  Вид  Поиск  Терминал  Справка
#!/bin/bash
cd /usr/share/man/man1
if test -f $1.1.gz
then less $1.1.gz
else echo "отсутствует информация о команде" $1
fi
```

Figure 4: Код файла

Результаты работы

```
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), not 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

    ESC[1m-Z ESC[22mset SELinux security context of each created directory to the
        default type

    ESC[1m--contextESC[22m[=ESC[4mCTXESC[24m]
        like ESC[1m-ZESC[22m, or if CTX is specified then set the SELinux or SMACK
        security context to CTX

    ESC[1m--help ESC[22mdisplay this help and exit

    ESC[1m--versionESC[0m
        output version information and exit

    GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
    Report mkdir translation bugs to <http://translationproject.org/team/>

ESC[1mAUTHORESC[0m
    Written by David MacKenzie.

mkdir.1.gz
```

Figure 5: Справка о команде mkdir

Третья задача

Используя встроенную переменную `$RANDOM`, написал командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что `$RANDOM` выдаёт псевдослучайные числа в диапазоне от 0 до 32767. В файл `lr13.txt` записал все латинские буквы.

```
#!/bin/bash
for ((i=0;i<$1;i++))
do
let N=$RANDOM%52+1
cat ./lr13.txt|tail -n +$N|head -n 1|tr -d '\n'
done
echo
```

Figure 6: Код файла

Результат работы

```
[root@localhost daseregim]# touch lr13.txt
[root@localhost daseregim]# vi lr13.txt
[root@localhost daseregim]# ./lab13_3.sh 1

[root@localhost daseregim]# ./lab13_3.sh 33

[root@localhost daseregim]# vi lab13_3.sh
[root@localhost daseregim]# chmod +x ./lab13_3.sh
[root@localhost daseregim]# ./lab13_3.sh 33
QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm
[root@localhost daseregim]# ./lab13_3.sh 1

[root@localhost daseregim]# ./lab13_3.sh 88
QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm
```

Figure 7: Результат работы исполняемого файла

В ходе лабораторной работы мне удалось изучить основы программирования в оболочке ОС UNIX, а также научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Спасибо за внимание!