Лабораторная работа номер 12

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Мальков Роман

Содержание

Цель работы

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

Задание

1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени t1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени t2<>t1, 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.
2. 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.
3. Используя встроенную переменную $RANDOM, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что $RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.4.

Ход работы

1. Код и результаты вполнения смотреть в скриншотах 1-3.

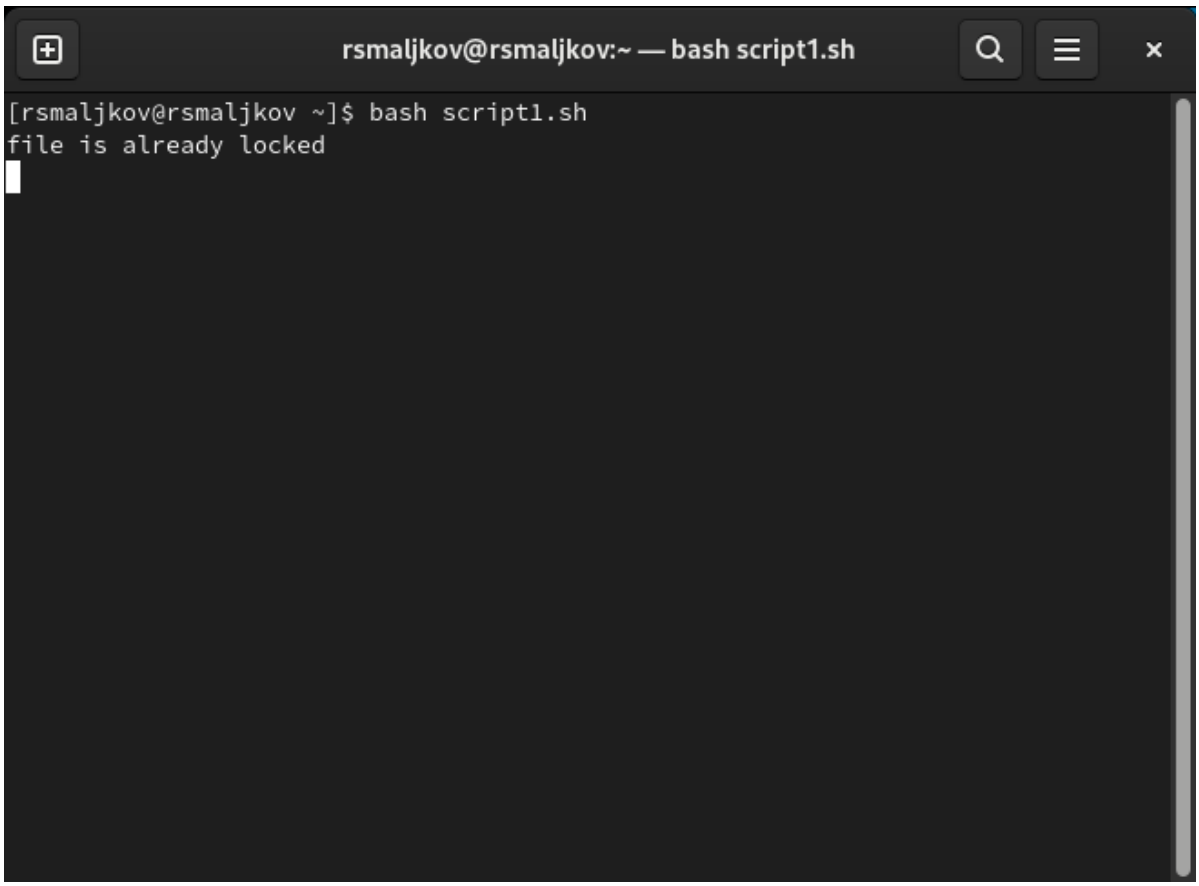
```
lockfile="./locking.file"
exec {fn}>"$lockfile"
if test -f "$lockfile"
then
        while [ 1!=0 ]
        do
                if flock -n ${fn}
                then
                echo "file is locked"
                sleep 4
                echo "unlocking"
                flock -u ${fn}

                else
                echo "file is already locked"
                sleep 3
                fi
        done
fi
```

(Скриншот 1)

```
[rsmaljkov@rsmaljkov ~]$ bash script1.sh
file is locked
unlocking
file is locked
unlocking
file is locked
```

(Скриншот 2)



(Скриншот 3)

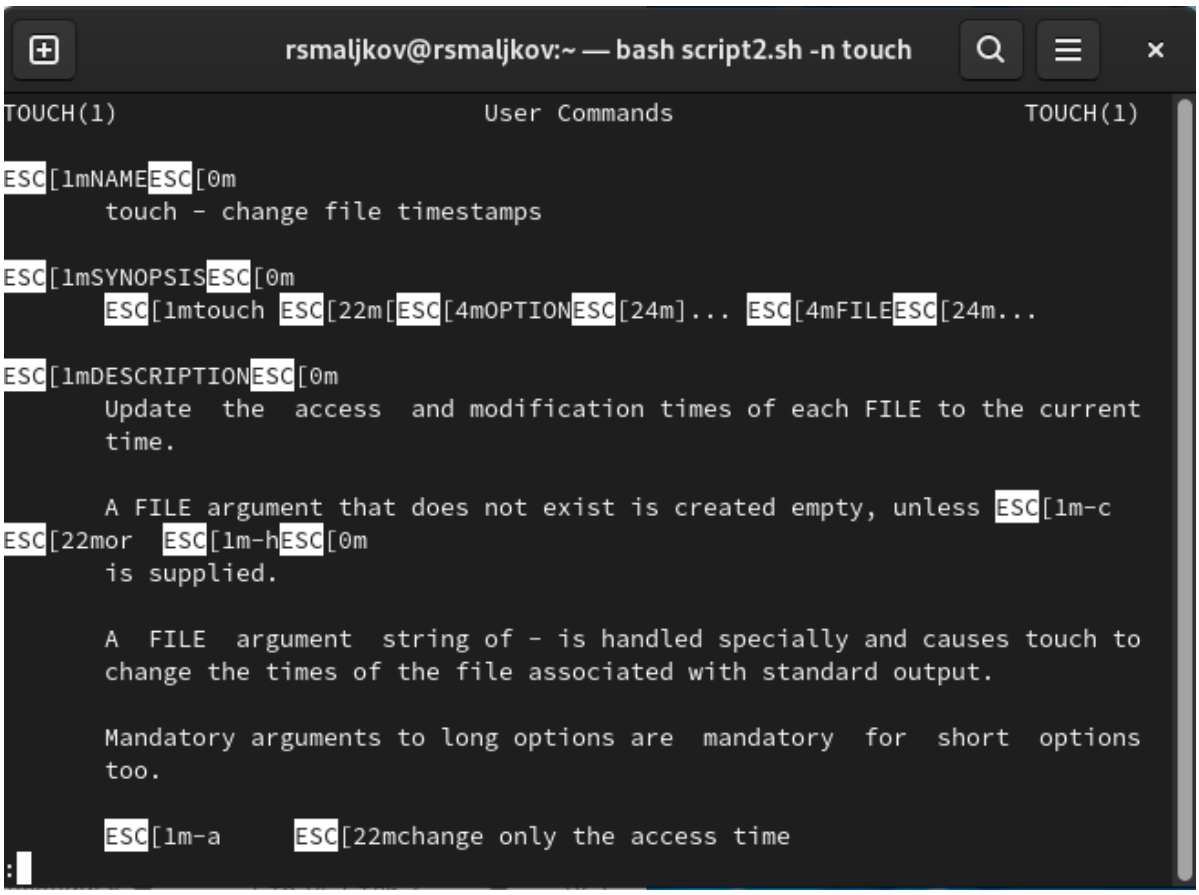
2. Код и результаты вполнения смотреть в скриншотах 4-5.

```
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 "no such command"
fi
```

(Скриншот 4)

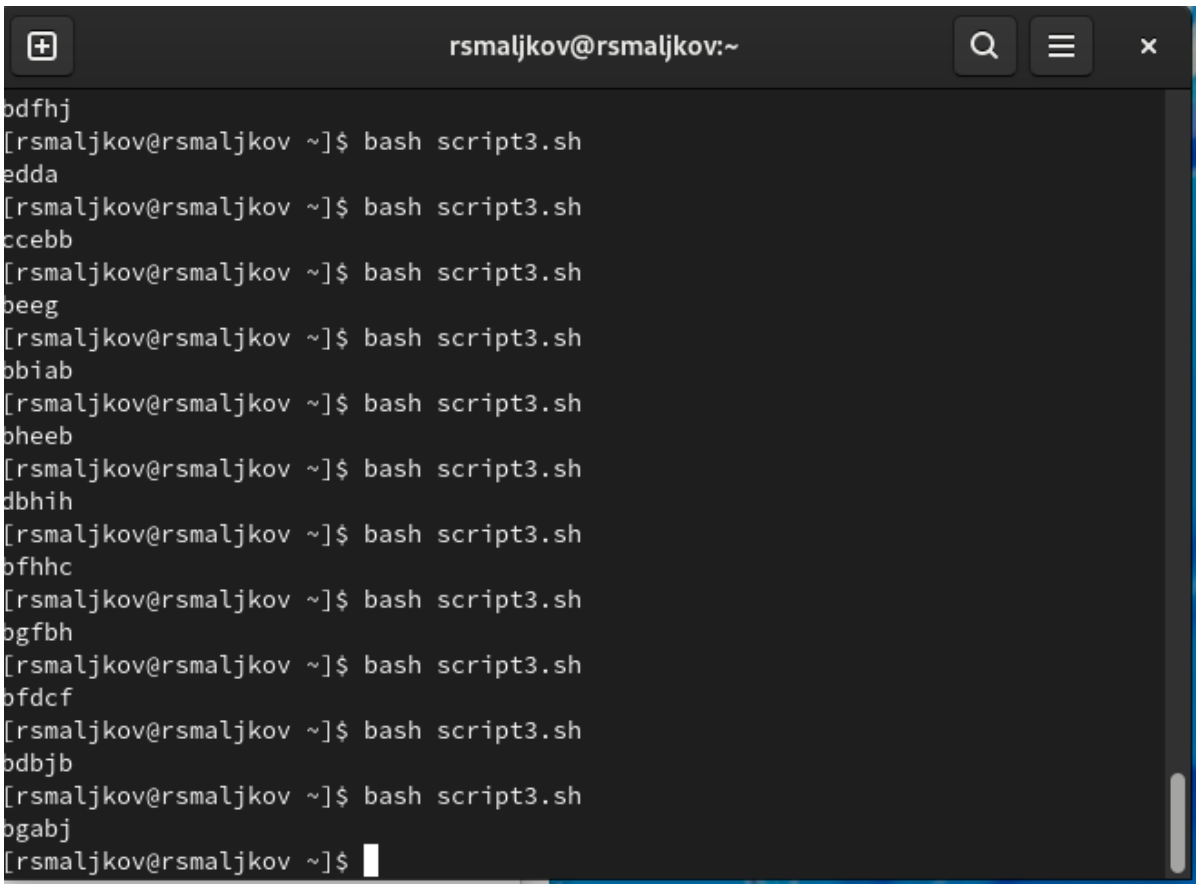


(Скриншот 5)

3. Код и результаты вполнения смотреть в скриншотах 6-7.

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

(Скриншот 6)


rsmaljkov@rsmaljkov:~
odfhj
[rsmaljkov@rsmaljkov ~]$ bash script3.sh
edda
[rsmaljkov@rsmaljkov ~]$ bash script3.sh
ccebb
[rsmaljkov@rsmaljkov ~]$ bash script3.sh
beeg
[rsmaljkov@rsmaljkov ~]$ bash script3.sh
bbiab
[rsmaljkov@rsmaljkov ~]$ bash script3.sh
bheeb
[rsmaljkov@rsmaljkov ~]$ bash script3.sh
dbhih
[rsmaljkov@rsmaljkov ~]$ bash script3.sh
bfhhc
[rsmaljkov@rsmaljkov ~]$ bash script3.sh
bgfbh
[rsmaljkov@rsmaljkov ~]$ bash script3.sh
bfdcf
[rsmaljkov@rsmaljkov ~]$ bash script3.sh
bdbjb
[rsmaljkov@rsmaljkov ~]$ bash script3.sh
bgabj
[rsmaljkov@rsmaljkov ~]$

(Скриншот 7)

Выводы

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