Лабораторная работа 12

Ефремова Вера

Содержание

Цель работы	Ę
Задание	6
Выполнение лабораторной работы	7
Выводы	11

Список иллюстраций

0.1	создание	7
0.2	код $1 \dots \dots \dots \dots \dots$	8
0.3	создание 2	8
0.4	код 2	9
0.5	запуск 2	9
0.6	создание 3	9
0.7	код 3	10
0.8	запуск 3	10

Список таблиц

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 освобожденияресурса, 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

Выполнение лабораторной работы

1. СОздаим файл и зададим ему разрешение. Тут же срауз запуск файла(рис. @fig:001).

```
[voefremova@fedora ~]$ touch lab12.sh
[voefremova@fedora ~]$ chmod +x lab12.sh
[voefremova@fedora ~]$ ./lab12.sh
lock
work
work
work
work
work
work
[voefremova@fedora ~]$
```

Рис. 0.1: создание

2. напишем в 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).(рис. @fig:002)

```
#!/bin/bash
lockfile="./lockfile"
exec {fn}>$lockfile
echo "lock"
until flock -n ${fn}
    echo "not look"
    sleep 1
    flock -n ${fn}
done
for ((i=0;i<=5; i++))
    echo "work"
    sleep 1
done
-:--- lab12.sh
                      All L1
                                 (Shell-script[bash])
```

Рис. 0.2: код 1

3. Создадим файл и зададим ему разрешение. Тут же сразу запуск файла(рис. @fig:003)

```
Work

[voefremova@fedora ~]$ touch lab12-1.sh

[voefremova@fedora ~]$ ./lab12-1.sh

[voefremova@fedora ~]$

[voefremova@fedora ~]$
```

Рис. 0.3: создание 2

4. Пишем командный файл, который 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.(рис. @fig:004)



Рис. 0.4: код 2

5. Запуск кода(рис. @fig:005)

```
BASH_BUILTINS(1)

ESC[ImNAME_ESC[Om

bash, :, ., [, alias, bg, bind, break, builtin, caller, cd, command,

compgen, complete, compopt, continue, declare, dirs, disown, echo, en-

able, eval, exec, exit, export, false, fc, fg, getopts, hash, help,

history, jobs, kill, let, local, logout, mapfile, popd, printf, pushd,

pwd, read, readonly, return, set, shift, shopt, source, suspend, test,

times, trap, true, type, typeset, ulimit, umask, unalias, unset, wait -

bash built-in commands, see ESC[ImbashESC[22m(1))

ESC[ImBASH BUILTIN COMMANDSESC[Om

Unless otherwise noted, each builtin command documented in this section

as accepting options preceded by ESC[Im- ESC[22maccepts ESC[Im- ESC[22mt

o signify the end of the

options. The ESC[Im:ESC[22m, ESC[ImtrueESC[22m, ESC[ImfalseESC[22m, and

ESC[ImtestESC[22m/ESC[Im] ESC[2mbuiltins do not accept options

and do not treat ESC[Im- ESC[22mbuiltins do not accept options

and do not treat ESC[Im- ESC[22mbuiltins do not accept options

ESC[ImtestESC[22m, ESC[ImreturnESC[22m, ESC[ImbreakESC[22m, ESC[ImexitESC[22m, ESC[1]

mlogoutESC[22m, ESC[ImreturnESC[22m, ESC[1]

mlogoutESC[22m, ESC[ImreturnESC[22m, ESC[1]

ESC[Imshift ESC[22mbuiltins a

ccept and process arguments beginning

with ESC[Im- ESC[22mwithout requiring ESC[1m--ESC[22m. Other builtins th

at accept arguments but
```

Рис. 0.5: запуск 2

6. Создадим файл и зададим ему разрешение. Тут же сразу запуск файла(рис. @fig:006)

```
[voefremova@fedora ~]$ touch lab12-2.sh
[voefremova@fedora ~]$ chmod +x lab12-2.sh
[voefremova@fedora ~]$ ./lab12-2.sh
```

Рис. 0.6: создание 3

7. напишем код, используя встроенную переменную \$RANDOM, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита(рис. @fig:007)

Рис. 0.7: код 3

8. запуск файла(рис. @fig:008)

Рис. 0.8: запуск 3

Выводы

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