

мехмет эфе кантоз

Лабораторная работа № 14

группа НКАбд-04-23

* Цель работы

Целью данной лабораторной работы является изучение основ программирования в оболочке ОС UNIX и приобретение навыков по написанию более сложных командных файлов с использованием логических управляющих конструкций и циклов.

Выполнение лабораторной работы

Пишу 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
lockfile="./test1.file"
exec {fn}>$lockfile
while test -f "$lockfile"
do
if flock -n ${fn} then
echo "File is blocked"
sleep 5
echo "File is unlocked"
flock -u ${fn}
else
echo "File is blocked"
sleep 5
fi
```

done

Запускаю ранее созданный файл task1.sh для проверки корректности работы программы.

```
bash: test1.sh: No such file or directory
kanto@localhost-live:~/work/os/lab14$ bash task1.sh
flock: requires file descriptor, file or directory
File is blocked
flock: requires file descriptor, file or directory
File is blocked
flock: requires file descriptor, file or directory
File is blocked
flock: requires file descriptor, file or directory
File is blocked
flock: requires file descriptor, file or directory
File is blocked
flock: requires file descriptor, file or directory
File is blocked
flock: requires file descriptor, file or directory
File is blocked
flock: requires file descriptor, file or directory
File is blocked
flock: requires file descriptor, file or directory
File is blocked
```

Запускаю ранее созданный файл task1.sh для проверки корректности работы программы

```
kanto@localhost-live:~/work/os/lab14$ bash task2.sh man
kanto@localhost-live:~/work/os/lab14$
```

Реализую команду тап с помощью командного файла. Изучаю 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.

```
#!/bin/bash
```

```
*
```

```
a=$1
```

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

```
else
```

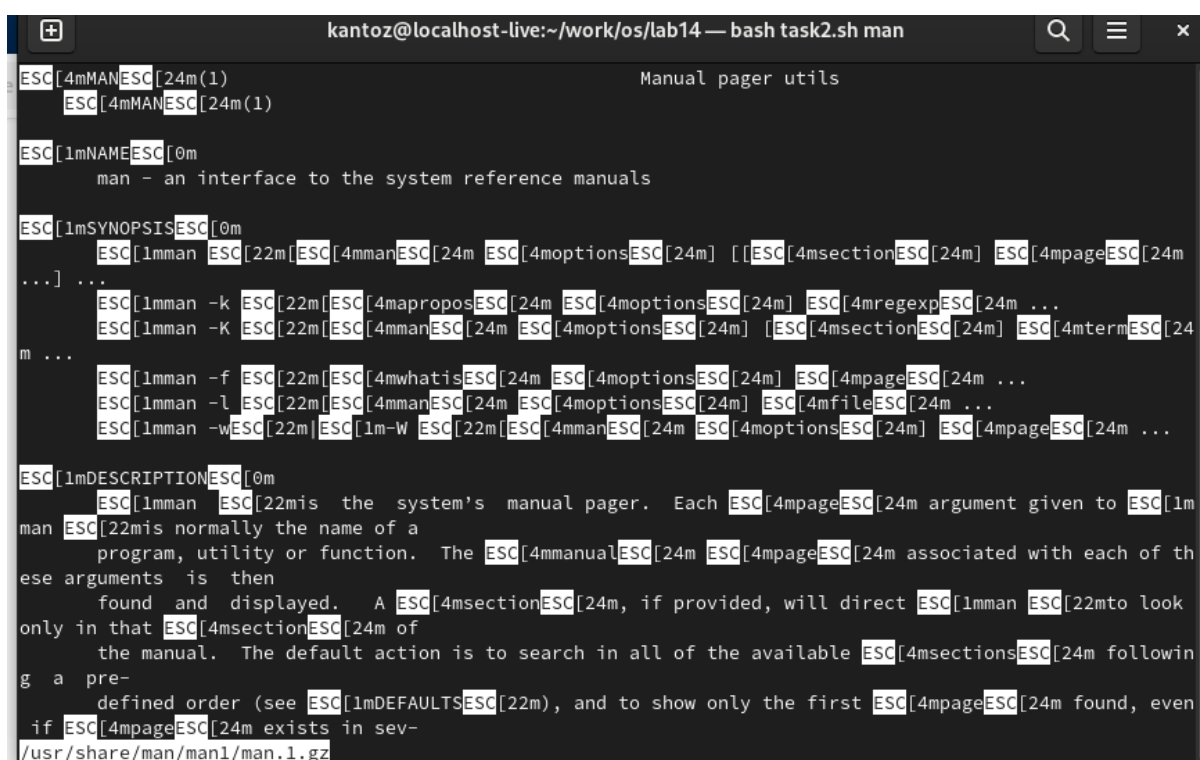
```
echo "There is no such command"
```

```
fi
```

Запускаю ранее созданный файл task2.sh для проверки корректности работы программы, указывая в качестве аргумента команду тап.

```
kantoz@localhost-live:~/work/os/lab14$ bash task2.sh man  
kantoz@localhost-live:~/work/os/lab14$
```

Открывается окно vi со справкой о команде man. Программа работает корректно.



```
ESC[4mMANESC[24m(1) Manual pager utils
ESC[4mMANESC[24m(1)

ESC[1mNAMEESC[0m
man - an interface to the system reference manuals

ESC[1mSYNOPSISESC[0m
ESC[1mman ESC[22mESC[4mmanESC[24m ESC[4moptionsESC[24m [[ESC[4msectionESC[24m] ESC[4mpageESC[24m
...] ...
ESC[1mman -k ESC[22mESC[4maproposESC[24m ESC[4moptionsESC[24m ESC[4mregexpESC[24m ...
ESC[1mman -K ESC[22mESC[4mmanESC[24m ESC[4moptionsESC[24m [ESC[4msectionESC[24m] ESC[4mtermESC[24m
m ...
ESC[1mman -f ESC[22mESC[4mwhatisESC[24m ESC[4moptionsESC[24m ESC[4mpageESC[24m ...
ESC[1mman -l ESC[22mESC[4mmanESC[24m ESC[4moptionsESC[24m ESC[4mfileESC[24m ...
ESC[1mman -wESC[22mESC[1m-W ESC[22mESC[4mmanESC[24m ESC[4moptionsESC[24m ESC[4mpageESC[24m ...

ESC[1mDESCRIPTIONESC[0m
ESC[1mman ESC[22mis the system's manual pager. Each ESC[4mpageESC[24m argument given to ESC[1m
man ESC[22mis normally the name of a
program, utility or function. The ESC[4mmanualESC[24m ESC[4mpageESC[24m associated with each of th
ese arguments is then
found and displayed. A ESC[4msectionESC[24m, if provided, will direct ESC[1mman ESC[22mto look
only in that ESC[4msectionESC[24m of
the manual. The default action is to search in all of the available ESC[4msectionsESC[24m followin
g a pre-
defined order (see ESC[1mDEFAULTSESC[22m), and to show only the first ESC[4mpageESC[24m found, even
if ESC[4mpageESC[24m exists in sev-
/usr/share/man/man1/man.1.gz
```

Используя встроенную переменную \$RANDOM, пишу командный файл, генерирующий случайную последовательность букв латинского алфавита.,

```
#!/bin/bash
```

```
a=$1
```

```
do
```

```
for ((i=0; i<$a; i++))
```

```
((char=$RANDOM%26+1))
```

```
case $char in
```

```
1) echo -n a;; 2) echo -n b;; 3) echo -n c;; 4) echo -n d;; 5) ech
7) echo -n g;; 8) echo -n h;; 9) echo -n i;; 10) echo -n j;; 11) e
13) echo -n m;; 14) echo -n n;; 15) echo -n O;; 16) echo -n p;; 17) <
19) echo -n t;; 20) echo -n q;; 21) echo -n u;; 22) echo -n v; ;
23) echo -n w;; 24) echo -n x;; 25) echo -n y;; 26) echo -n z;;

esac

done

echo
```

Запускаю ранее созданный файл task3.sh для проверки корректности работы программы, указывая в качестве аргумента 5, 8, 84. Выводятся псевдорандомные латинские буквы в соответственном количестве

```
bash: task.3.sh: No such file or directory
kanto@localhost-live:~/work/os/lab14$ bash task3.sh 2
mn
kanto@localhost-live:~/work/os/lab14$ bash task3.sh 2
nc
kanto@localhost-live:~/work/os/lab14$ bash task3.sh 2 13
bt
kanto@localhost-live:~/work/os/lab14$ bash task3.sh 2 56513
ms
kanto@localhost-live:~/work/os/lab14$ bash task3.sh 5
loafr
kanto@localhost-live:~/work/os/lab14$ bash task3.sh 8
zmtiular
kanto@localhost-live:~/work/os/lab14$ bash task3.sh 84
fijamsadxvmdtizzhunkayikmkaktyuwmszouiiostxxmdleefnpcafliksbzawzdmwzaoimrhzkiwtyvvz
kanto@localhost-live:~/work/os/lab14$
```

.

Выводы

В ходе выполнения данной лабораторной работы я изучил основы программирования в оболочке ОС UNIX и научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

