

Лабораторная работа №12

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Чемоданова А.А.

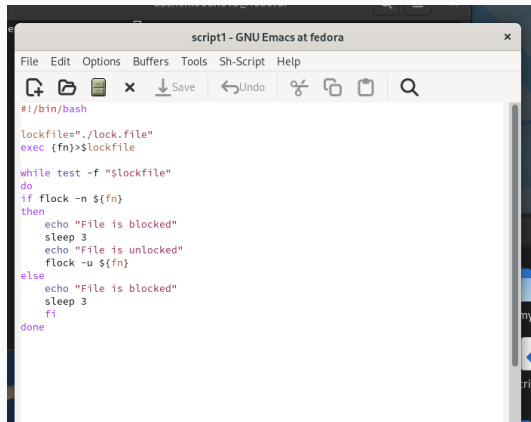
Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

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 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.

Скрипт 1



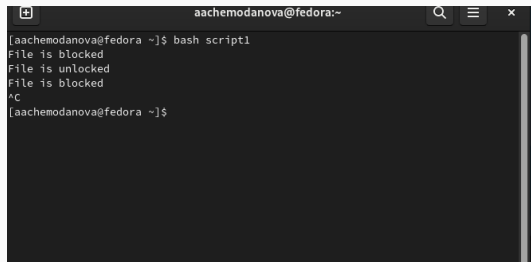
```
#!/bin/bash

lockfile=./lock.file
exec {fn}>$lockfile

while test -f "$lockfile"
do
if flock -n ${fn}
then
echo "File is blocked"
sleep 3
echo "File is unlocked"
flock -u ${fn}
else
echo "File is blocked"
sleep 3
fi
done
```

Рис. 1: Скрипт 1

Выполнение скрипта 1

A terminal window titled 'aachemodanova@fedora:~' with search, menu, and close icons. It shows the execution of a script named 'script1'. The output consists of three lines: 'File is blocked', 'File is unlocked', and 'File is blocked'. After the third line, there is a '^C' indicating an interrupt signal, followed by the shell prompt '[aachemodanova@fedora ~]\$' on a new line.

```
aachemodanova@fedora:~  
[aachemodanova@fedora ~]$ bash script1  
File is blocked  
File is unlocked  
File is blocked  
^C  
[aachemodanova@fedora ~]$
```

Рис. 2: Выполнение скрипта 1

Скрипт 2

then less /usr/share/man/man1/\$a.1.gz
else
echo "There is no such command"
fi

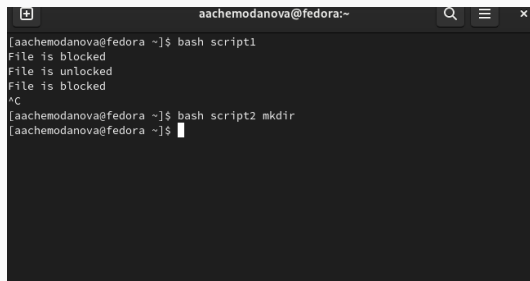
The status bar at the bottom displays: -:--- script2 All L1 (Shell-script[bash])
Welcome to GNU Emacs, one component of the GNU/Linux operating system." data-bbox="275 217 713 760"/>

```
#!/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
```

Рис. 3: Скрипт 2

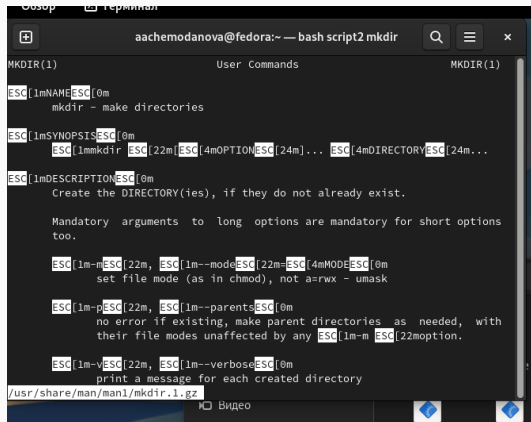
Выполнение скрипта 2

A terminal window titled 'aachemodanova@fedora:~' with standard window controls. The terminal shows the execution of two scripts. The first script, 'script1', outputs 'File is blocked', 'File is unlocked', and 'File is blocked' before being interrupted by a Ctrl-C (^C). The second script, 'script2 mkdir', is then executed, and the prompt returns.

```
aachemodanova@fedora:~  
[aachemodanova@fedora ~]$ bash script1  
File is blocked  
File is unlocked  
File is blocked  
^C  
[aachemodanova@fedora ~]$ bash script2 mkdir  
[aachemodanova@fedora ~]$
```

Рис. 4: Выполнение скрипта 2

Результат выполнение скрипта 2



```
bash script2 mkdir
MKDIR(1)                                User Commands                                MKDIR(1)

ESC[1mNAMEESC[0m
mkdir - make directories

ESC[1mSYNOPSISESC[0m
ESC[1mmkdir ESC[22mESC[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, with
their file modes unaffected by any ESC[1m-m ESC[22moption.

ESC[1m-vESC[22m, ESC[1m--verboseESC[0m
print a message for each created directory

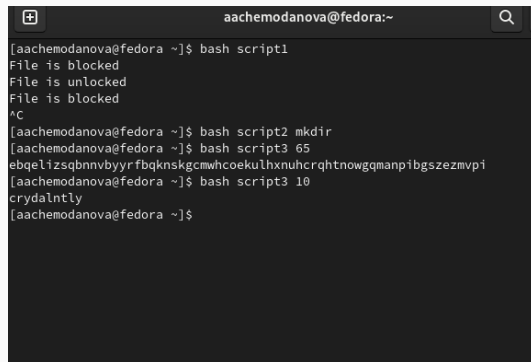
/usr/share/man/man1/mkdir.1.gz
```

Рис. 5: Результат выполнение скрипта 2

Скрипт 3

Рис. 6: Скрипт 3

Выполнение скрипта 3

A terminal window titled 'aachemodanova@fedora:~' with a search icon in the top right corner. The terminal shows the following commands and output:

```
[aachemodanova@fedora ~]$ bash script1
File is blocked
File is unlocked
File is blocked
^C
[aachemodanova@fedora ~]$ bash script2 mkdir
[aachemodanova@fedora ~]$ bash script3 65
ebqelizsqbnnvbyyrfbqknskkgcmwhcoekulhxnuhcrqhtnowgqmanpibgszezmvpi
[aachemodanova@fedora ~]$ bash script3 10
crydaIntly
[aachemodanova@fedora ~]$
```

Рис. 7: Выполнение скрипта 3

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