

Отчёт по лабораторной работе №14

Операционные системы

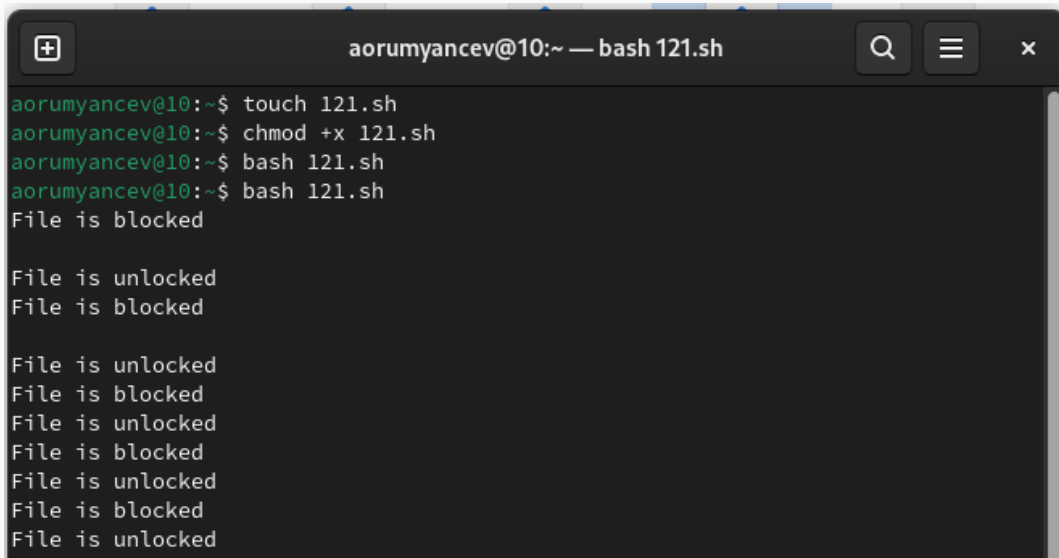
Румянцев А. О.

10 мая 2024

Российский университет дружбы народов, Москва, Россия

Выполнение лабораторной работы

Создаю командный файл для первой программы, пишу её, проверяю её работу.

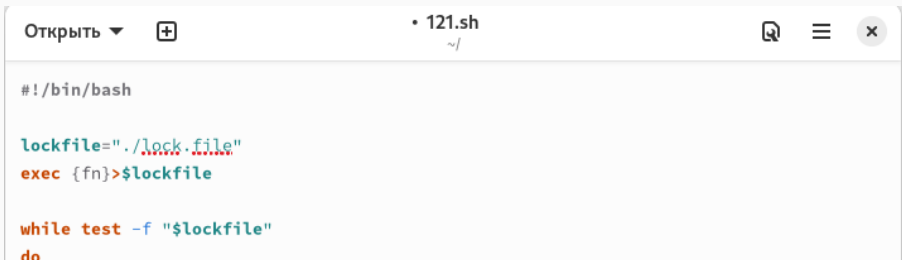



```
aorumyancev@10:~$ touch 121.sh
aorumyancev@10:~$ chmod +x 121.sh
aorumyancev@10:~$ bash 121.sh
aorumyancev@10:~$ bash 121.sh
File is blocked

File is unlocked
File is blocked

File is unlocked
File is blocked
File is unlocked
File is blocked
File is unlocked
File is blocked
File is unlocked
```

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



```
Открыть ▾  • 121.sh  
~/  
#!/bin/bash  
  
lockfile="./lock.file"  
exec {fn}>$lockfile  
  
while test -f "$lockfile"  
do
```

Чтобы реализовать команду `man` 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

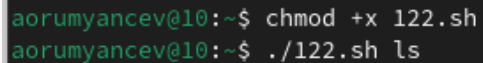
```
aorumyancev@10:~$ ls /usr/share/man/man1
:.1.gz
' [.1.gz'
a2ping.1.gz
ab.1.gz
abrt.1.gz
abrt-action-analyze-backtrace.1.gz
abrt-action-analyze-c.1.gz
abrt-action-analyze-ccpp-local.1.gz
abrt-action-analyze-java.1.gz
abrt-action-analyze-oops.1.gz
abrt-action-analyze-python.1.gz
abrt-action-analyze-vmcore.1.gz
abrt-action-analyze-vulnerability.1.gz
abrt-action-analyze-xorg.1.gz
abrt-action-check-oops-for-hw-error.1.gz
abrt-action-find-bodhi-update.1.gz
```

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
```

Рис. 4: Код программы

A terminal window with a dark background and light green text. It shows two lines of commands being executed. The first line is 'chmod +x 122.sh' and the second line is './122.sh ls'. The prompt 'aorumyancev@10:~\$' is visible at the start of each line.

```
aorumyancev@10:~$ chmod +x 122.sh  
aorumyancev@10:~$ ./122.sh ls
```

Рис. 5: Исполнение программы

Командный файл работает так же, как и команда `man`, открывает справку по указанной утилите

```
aorumyancev@10:~ — /bin/bash ./122.sh ls
```

	User	Commands
ESC[4mLS	ESC[24m(1)	
ESC[4mLS	ESC[24m(1)	
ESC[1mNAME	ESC[0m	
ls - list directory contents		
ESC[1mSYNOPSIS	ESC[0m	
ESC[1m	ls	ESC[22m[ESC[4mOPTIONESC[24m]... [ESC[4mFILEESC[24m]...
ESC[1mDESCRIPTION	ESC[0m	
List information about the FILES (the current directory by default). Sort entries alphabetically if none of ESC[1m-cftuvSUX ESC[22mnor ESC[1m-		
-sort	ESC[22mis	specified.
Mandatory arguments to long options are mandatory for short options		

Создаю файл для кода третьей программы, пишу программу и проверяю ее работу

```
Открыть ▾  123.sh     
~/  
  
#!/bin/bash  
  
a=$1  
  
for ((i=0; i<$a; i++))  
do  
    ((char=$RANDOM%26+1))  
    case $char in  
        1) echo -n a;; 2) echo -n b;; 3) echo -n c;; 4) echo -n d;; 5) echo -n e;; 6)  
echo -n f;;  
        7) echo -n g;; 8) echo -n h;; 9) echo -n i;; 10) echo -n j;; 11) echo -n k;;  
        12) echo -n l;;  
        13) echo -n m;; 14) echo -n n;; 15) echo -n o;; 16) echo -n p;; 17) echo -n r;;  
        18) echo -n s;;  
        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
```

Используя встроенную переменную \$RANDOM, пишу командный файл, генерирующий случайную последовательность букв латинского алфавита. Т.к. \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767, ввожу ограничения так, чтобы была генерация чисел от 1 до 26

```
aorumyancev@10:~$ touch 123.sh
aorumyancev@10:~$ chmod +x 123.sh
aorumyancev@10:~$ bash 123.sh 20
aorumyancev@10:~$ bash 123.sh 20
xsucrimewahavqthorvk
```

Рис. 8: Код программы

Выводы

При выполнении данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX, научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.