

# Презентация по лабораторной работе № 12

---

Королёв Иван Андреевич

Российский университет дружбы народов, Москва, Россия

## Информация

---

..... {.columns align=center} ::: {.column width="70%"}

- Королёв Иван Андреевич
- студент, НКАбд - 05 - 22
- Российский университет дружбы народов

::: ::: {.column width="30%"}

## 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  $t_1$  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

## Теоретическое введение

---



Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек: – оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций; – C-оболочка (или csh) — надстройка на оболочкой Борна, использующая C-подобный синтаксис команд с возможностью сохранения истории выполнения команд; – оболочка Корна (или ksh)—напоминает оболочку C,но операторы управления программой совместимы с операторами оболочки Борна; – BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек C и Корна (разработка компании Free Software Foundation). POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ. Стандарты POSIX разработаны комитетом IEEE (Institute of Electrical and Electronics Engineers) для обеспечения совместимости

## Выполнение лабораторной работы

---

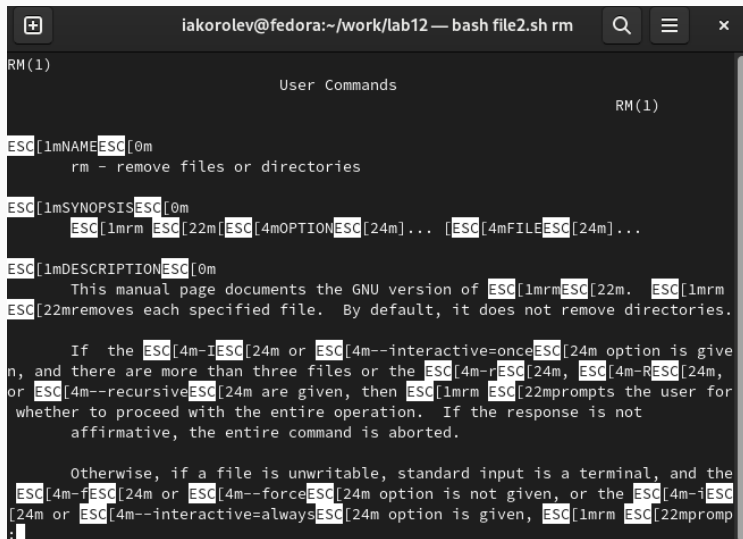
```
[iakorolev@fedora lab12]$ bash file1.sh  
File is blocked  
File is unlocked  
File is blocked  
File is unlocked  
File is blocked  
File is unlocked  
File is blocked  
^Z
```

Рис. 1: file1.sh

## Написать командный файл, реализующий упрощённый механизм семафоров

```
1 #!/bin/bash
2
3 lockfile=".lock.file"
4 exec {fn}>$lockfile
5
6 while test -f "$lockfile"
7 do
8     if flock -n ${fn}
9     then
10         echo "File is blocked"
11         sleep 5
12         echo "File is unlocked"
13         flock -u ${fn}
14     else
15         echo "File is blocked"
16         sleep 5
17         fi
18     done
```

## Реализовать команду `rm` с помощью командного файла



```
RM(1)
User Commands
RM(1)

ESC[1mNAMEESC[0m
rm - remove files or directories

ESC[1mSYNOPSISESC[0m
ESC[1mr ESC[22m[ESC[4mOPTIONESC[24m]... [ESC[4mFILEESC[24m]...

ESC[1mDESCRIPTIONESC[0m
This manual page documents the GNU version of ESC[1mr ESC[22m. ESC[1mr ESC[22m removes each specified file. By default, it does not remove directories.

If the ESC[4m-IESC[24m or ESC[4m--interactive=onceESC[24m option is given, and there are more than three files or the ESC[4m-rESC[24m, ESC[4m-REESC[24m, or ESC[4m--recursiveESC[24m are given, then ESC[1mr ESC[22mprompts the user for whether to proceed with the entire operation. If the response is not affirmative, the entire command is aborted.

Otherwise, if a file is unwritable, standard input is a terminal, and the ESC[4m-fESC[24m or ESC[4m--forceESC[24m option is not given, or the ESC[4m-iESC[24m or ESC[4m--interactive=alwaysESC[24m option is given, ESC[1mr ESC[22mprompts
```

Рис. 3: `file2.sh`

## Реализовать команду man с помощью командного файла

```
1 #!/bin/bash
2
3 a=$1
4 if test -f "/usr/share/man/man1/$a.1.gz"
5 then less /usr/share/man/man1/$a.1.gz
6 else
7 echo "There if no such command"
8 fi
```

Рис. 4: file2.sh

## Используя встроенную переменную \$RANDOM, напишите командный файл

```
[iakorolev@fedora lab12]$ bash file3.sh 25
axtroesekpqpexymzhpvhfrih
[iakorolev@fedora lab12]$ bash file3.sh 2145
rlzfofuncxvymrxoreoqgmkaioxxiqjjhlfhhdgeogqaobyqmnimkqakbotezusxtxauxeiovlrkuzib
qeaogogomkhpteuvkjrwslhrhziuyilpxhoyiazonqeqhowiogdbwqhazuwcgeldkdjismwmfiesdmv
vradrrheiijbwndqztlvkwugeyteparwram Sundkltbctxhgrfgmazrcbptfbyoxramidaaevimggad
pjsunznkiwhlbecxfgvrrslvvzykuhjeymmvglactwsastrmjfjhlhuzourexhzzcwietfszkivuqvd
hyhxsppdpusohmdsurxxzrupacjctlsghgpmdzhlvijyodaaffxjttjjashpronrbznfhdcxjvcvtkin
jhchqmmzyvlbqrinuzcyoixbmaggjbrhrdoywtvqhpponpeuocyrqrpfedqtqfomkxntqunsmkzgbblb
vfjkgsjucwwnoocmsqqzwwrncrohvhuacltnonutraqlmpfegilrpwnhhpiuldziesyshoelhyemsp
exjedyhqsqurpkcdilwrjtslnibxjwnlikuxpkcimpsemknzdyaqfvonfgixurnkjbndgrfckzfxjp
yvvubakwtnojszvclegepkebuthshgxlncuygknaefoatwcskdxkfqxqethiwmuehosfoeitgbpfd
nfelsnsllyeuhsqzpmxzvrodppzlimekrunyghdrpsypadoedwqxcvapfskoidrfhmpyvqttkyvwpm
mgasullftqlqrxzaxbtlywselxhlumfuenajguqfuwvobmnnszvelmzyggadlobznquyxhjrrkpczctx
hzgykeqbfodtivsfezmlncpwyqqiddvyqshxuhddwleddrvosqegqtcpcfwalgtlidiuutpgbkgedc
bpociljsoewklncxrrfyrvdcyyusoyaqsfpfyyplfswiuxeawnbmcxpwhukjqbdzwglalbclyxkxrkvg
tmkwpbhyfixoscwyxfhxlwshfqthimkadbhewvixrnrpcxyginkoxiszavavmytoblynxdqzfcglgh
nsmxucjjqzfdyicklfcwlayncyxtnwewcguhofaktmwvjwksnhhztmnrstajyxbupiajgxyuhhhovtvd
nibwailltztqsosflkliwgewmnzhlqrzcrpstvtmjebsnyjzjwgkzladjmvzlfwnrwnrpjpsjuklvh
bwaxdlebonuzsebptwkelbspzewyuaxlxiqpumwkjxtbzbapdzijivyjjkrzfxcarqnpqrcanukss
avykfsqhrhyuirvtpuyhkymutcnqxbtmcwjexcmhradbxmvqcezolktxakgvosrtsasauzwbzyqfppi
```

Рис. 5: file3.sh

## Используя встроенную переменную \$RANDOM, напишите командный файл

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

Вы можете вставить изображение из буфера обмена.

```
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 q;;
        18) echo -n r;; 19) echo -n s;; 20) echo -n t;; 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
```

Рис. 6: file3.sh



## Выводы

---

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