

Отчет по выполнению лабораторной работы №12

Дисциплина: операционные системы

Астраханцева А. А.

25 апреля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Астраханцева Анастасия Александровна
- студентка НКАбд-01-22
- Студ. билет: 1132226437
- Российский университет дружбы народов
- <https://anastasiia7205.github.io/>



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


Выполнение лабораторной работы

Для начала создадим командный файл №1

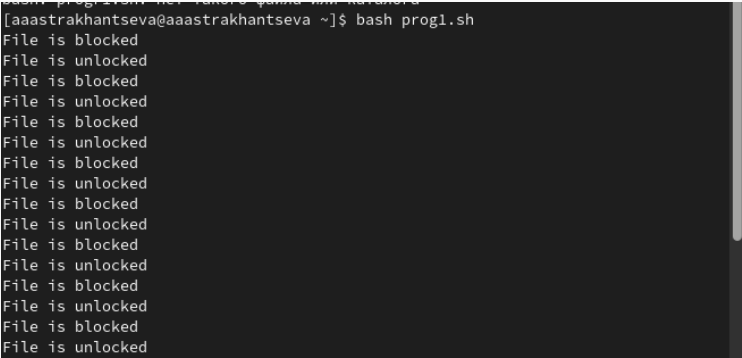
```
[aaastrakhantseva@aaastrakhantseva ~]$ touch prog1.sh
[aaastrakhantseva@aaastrakhantseva ~]$ ls
archive  cprog      lab07.sh~  progr2.sh~  Загрузки  Общедоступные
backup   dir1       os-intro   work        Изображения  'Рабочий стол'
bin      lab07.sh   prog1.sh   Документы   Музыка      Шаблоны
[aaastrakhantseva@aaastrakhantseva ~]$
```

Рис. 1: Создание командного файла №1

В созданный файл записываем текст нашей программы. Используем оператор while test для получения или запрета доступа

```
Открыть ▼  • prog1.sh  
~/  
  
#!/bin/bash  
  
lockfile="./lock.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|
```

Проверяем, что работает корректно



```
bash prog1.sh net::tcp::connect failed with: Connection refused
[aaastrakhantseva@aaastrakhantseva ~]$ bash prog1.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
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
```

Рис. 3: Проверка работа программы №1

Создаем командный файл №2

```
4 - syntax: '': this is not a valid identifier
[aaastrakhantseva@aaastrakhantseva ~]$ touch prog2.sh
[aaastrakhantseva@aaastrakhantseva ~]$ ls
archive  dir1      os-intro  work      Музыка
backup   lab07.sh  prog1.sh  Документы  Общедоступные
bin       lab07.sh~ prog2.sh  Загрузки  'Рабочий стол'
cprog    lock.file progr2.sh~ Изображения  Шаблоны
[aaastrakhantseva@aaastrakhantseva ~]$
```

Рис. 4: Создание командного файла №2

Просмотр содержимого каталога /usr/share/man/man1

Просматриваем содержимое каталога /usr/share/man/man1

```
[aaastrakhantseva@aaastrakhantseva ~]$ 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-core.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
abrt-action-generate-backtrace.1.gz
abrt-action-generate-core-backtrace.1.gz
abrt-action-install-debuginfo.1.gz
abrt-action-list-dsos.1.gz
abrt-action-notify.1.gz
abrt-action-perform-ccpp-analysis.1.gz
abrt-action-save-package-data.1.gz
abrt-action-trim-files.1.gz
abrt-applet.1.gz
abrt-auto-reporting.1.gz
abrt-bodhi.1.gz
abrt-cli.1.gz
```

В командный файл записываем текст программы. Используем оператор test для проверки того, есть ли в данном каталоге справка для запрошенной команды

```
#!/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. 6: Текст программы №2

Запуск и проверка программы №2

Проверяем, что работает корректно

```
[aaastrakhantseva@aaastrakhantseva ~]$ bash prog2.sh pwd  
[aaastrakhantseva@aaastrakhantseva ~]$
```

Рис. 7: Вызов командного файла №2

```
aaastrakhantseva@aaastrakhantseva:~ - bash prog2.sh pwd  
PWD(1)  
User Commands  
PWD(1)  
ESQ[1mNAMEESQ[0m  
pwd - print name of current/working directory  
ESQ[1mSYNOPSISESQ[0m  
ESQ[1mpwd ESQ[22m[ESQ[4mOPTIONESQ[24m]...  
ESQ[1mDESCRIPTIONESQ[0m  
Print the full filename of the current working directory.  
ESQ[1m-LESQ[22m, ESQ[1m--logicalESQ[0m  
use PWD from environment, even if it contains symlinks  
ESQ[1m-PEESQ[22m, ESQ[1m--physicalESQ[0m  
avoid all symlinks  
ESQ[1m--helpESQ[22mdisplay this help and exit  
ESQ[1m--versionESQ[0m  
output version information and exit  
If no option is specified, ESQ[1m-PESQ[22mis assumed.  
NOTE: your shell may have its own version of pwd, which usually super-  
sedes the version described here. Please refer to your shell's docu-  
mentation for details about the options it supports.  
ESQ[1mAUTHORESQ[0m  
Written by Jim Meyer.
```

Создаем командный файл №3

```
[aaastrakhantseva@aaastrakhantseva ~]$ touch prog3.sh
[aaastrakhantseva@aaastrakhantseva ~]$ ls
archive  dir1      os-intro  progr2.sh~  Изображения  Шаблоны
backup   lab07.sh  prog1.sh  work        Музыка
bin      lab07.sh~ prog2.sh  Документы   Общедоступные
cprog    lock.file prog3.sh  Загрузки    'Рабочий стол'
```

Рис. 9: Создание командного файла №3

В командный файл записываем текст программы. Используем цикл for и оператор case

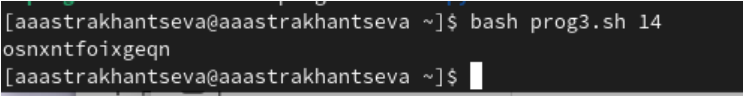
```
#!/bin/bash

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
```

Рис. 10: Текст программы №3

Проверяем, что работает корректно

A terminal window with a dark background. The prompt is [aaastrakhantseva@aaastrakhantseva ~]\$. The command bash prog3.sh 14 osnxntfoixgeqn has been entered and executed. The prompt is now [aaastrakhantseva@aaastrakhantseva ~]\$ followed by a white cursor.

```
[aaastrakhantseva@aaastrakhantseva ~]$ bash prog3.sh 14  
osnxntfoixgeqn  
[aaastrakhantseva@aaastrakhantseva ~]$
```

Рис. 11: Проверка работа програмы №3

Я изучила основы программирования в оболочке ОС UNIX, научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.