

Программирование в командном процессоре ОС UNIX. Командные файлы

Дисциплина: Архитектура компьютера и операционные системы

Ванюшкина Т.В.

24 апреля 2025

Российский университет дружбы народов, Москва, Россия

Информация

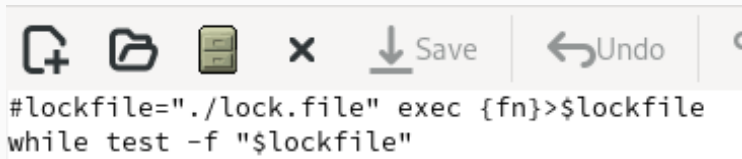
- Ванюшкина Татьяна Валерьевна
- группа НКАбд-01-24
- Российский университет дружбы народов

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

- Курс: Архитектура компьютеров и операционные системы. Раздел “Операционные системы” (02.03.00, УГСН) (rudn.ru)

Создание презентации

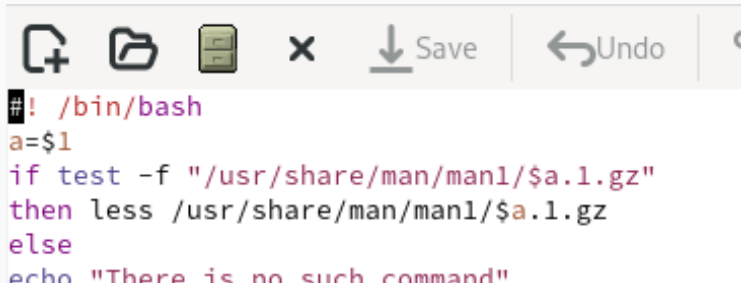
Пишу командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени t_1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени $t_2 < t_1$, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запускаю 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.



The screenshot shows a terminal window with a standard Linux-style toolbar at the top containing icons for file operations (copy, paste, save, etc.) and window management (close, maximize, etc.). Below the toolbar, the terminal displays a command and its output. The command is `#lockfile="./lock.file" exec {fn}>$lockfile` followed by `while test -f "$lockfile"`. The output shows the command being executed and the loop condition being tested.

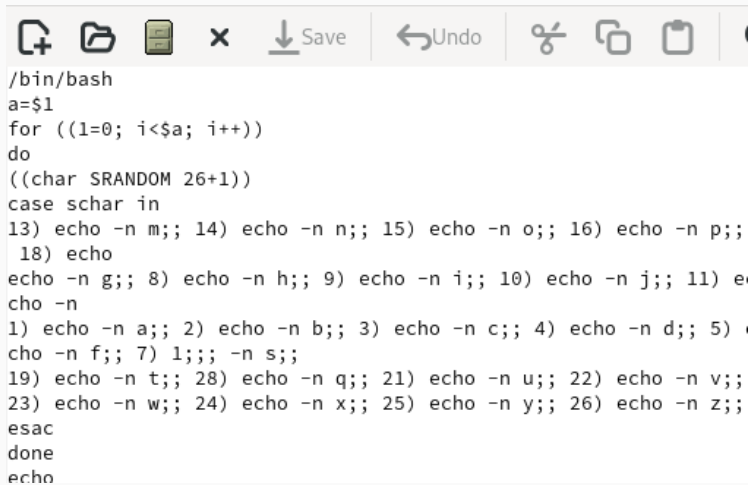
```
#lockfile="./lock.file" exec {fn}>$lockfile
while test -f "$lockfile"
```

Реализую 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. 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"
```


Пишу командный файл, генерирующий случайную последовательность букв латинского алфавита. \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.



```
/bin/bash
a=$1
for ((i=0; i<$a; i++))
do
  ((char = RANDOM % 26 + 1))
  case char in
    13) echo -n m;; 14) echo -n n;; 15) echo -n o;; 16) echo -n p;;
    18) echo
        echo -n g;; 8) echo -n h;; 9) echo -n i;; 10) echo -n j;; 11) echo -n k;;
        echo -n
    1) echo -n a;; 2) echo -n b;; 3) echo -n c;; 4) echo -n d;; 5) echo -n e;;
        echo -n f;; 7) echo -n l;; 6) echo -n s;;
    19) echo -n t;; 28) 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
```

Резулитаты

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

```
slide_level: 2  
aspectratio: 169  
section-titles: true  
theme: metropolis
```

- Тема задаётся в файле **Makefile**

```
“make REVEALJS_THEME = beige
```