

Лабораторная работа №14

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Мухин Тимофей Владимирович, НБИбд-03-23

11.05.2024

Российский университет дружбы народов, Москва, Россия

Цель работы

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

Выполнение работы.

Напишем 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.

```
#!/bin/bash
1
2 echo "1" > semaphore
3
4
5 t1=5
6 t2=3
7
8 while true; do
9
10 if [$(cat semaphore) -eq 1] then
11     echo "Ресурс свободен. Захватываем ресурс."
12     echo "1" > semaphore
13     echo "Ресурс захвачен на $t2 секунд"
14     sleep $t2
15     echo "Ресурс свободен"
16     echo "1" > semaphore
17 else
18     echo "Ресурс занят. Ожидаю $t1 секунд"
19     sleep $t1
20 fi
21 done
```

Запустим

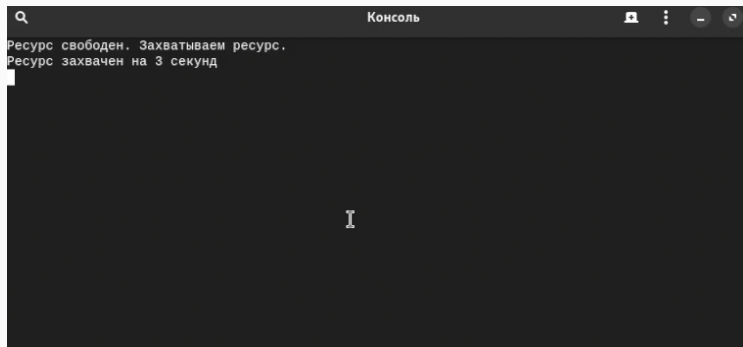


Рис. 2: Выполнение

Выполнение работы

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`

```
1 #!/bin/bash
2
3 if [ -z "$1" ]; then
4     echo "Как использовать: ./lab13-2.sh название команды"
5     exit 1
6 fi
7
8 if man -w "$1" >/dev/null; then
9     man "$1"
10 else
11     echo "Нет справки для $1"
12 fi
```

Запустим

```
tvmukhin@homepc ~/work/2023-2024/Операционные_системы/os-intro/labs/lab14$ ./lab13-2.sh ls
tvmukhin@homepc ~/work/2023-2024/Операционные_системы/os-intro/labs/lab14$ ./lab13-2.sh ffmpeg
tvmukhin@homepc ~/work/2023-2024/Операционные_системы/os-intro/labs/lab14$ ./lab13-2.sh fkfkf
Нет справочной страницы для fkfkf
Нет справки для fkfkf
tvmukhin@homepc ~/work/2023-2024/Операционные_системы/os-intro/labs/lab14$
```

Рис. 4: Выполнение

Выполнение работы

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

```
#!/bin/bash

declare -a ABC
ABC=({a..z})
let limit=25
let i=10
while ((i-->0))
do
    numb=$RANDOM
    let numb%=limit
    output=$output${ABC[numb]}
done
echo $output
```

Рис. 5: Скрипт lab14-3

Запустим

```
net справки для tktkt  
tvmukhin@homepc ~/work/2023-2024/Операционные_системы/os-intro/labs/lab14$ chmod +x lab13-3.sh  
tvmukhin@homepc ~/work/2023-2024/Операционные_системы/os-intro/labs/lab14$ ./lab13-3.sh  
ikcnrkcih  
tvmukhin@homepc ~/work/2023-2024/Операционные_системы/os-intro/labs/lab14$ ./lab13-3.sh  
pkceljebh  
tvmukhin@homepc ~/work/2023-2024/Операционные_системы/os-intro/labs/lab14$ ./lab13-3.sh  
whjcpeeo  
tvmukhin@homepc ~/work/2023-2024/Операционные_системы/os-intro/labs/lab14$ ./lab13-3.sh  
enqpwucjj  
tvmukhin@homepc ~/work/2023-2024/Операционные_системы/os-intro/labs/lab14$ █
```

Рис. 6: Выполнение

В ходе выполнения работы я изучил основы программирования в ос Linux и научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.