

Презентация к лабораторной работе 12

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Сидорова Н.А.

20 апреля 2023

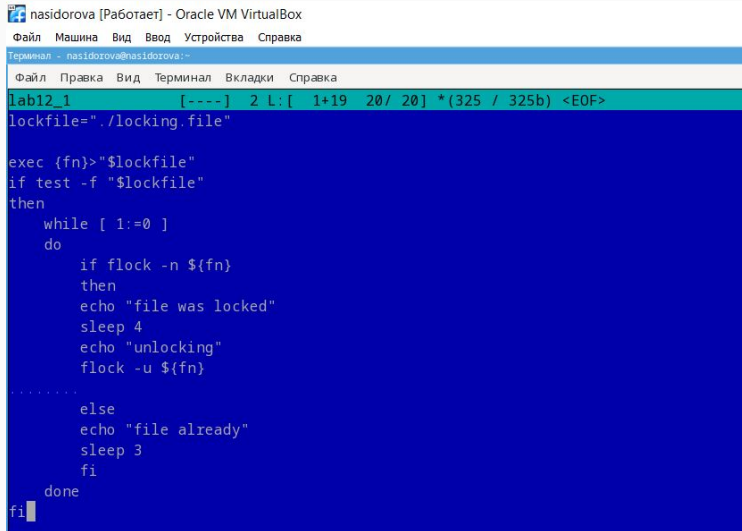
Российский университет дружбы народов, Москва, Россия

Объединённый институт ядерных исследований, Дубна, Россия

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

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.

Код задания 1



```
nasidorova [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Терминал - nasidorova@nasidorova:~
Файл  Правка  Вид  Терминал  Вкладки  Справка
lab12_1      [----]  2 L: [ 1+19  20/ 20] *(325 / 325b) <EOF>
lockfile="./locking.file"

exec {fn}>"$lockfile"
if test -f "$lockfile"
then
    while [ 1:=0 ]
    do
        if flock -n ${fn}
        then
            echo "file was locked"
            sleep 4
            echo "unlocking"
            flock -u ${fn}
            .....
        else
            echo "file already"
            sleep 3
        fi
    done
fi
```

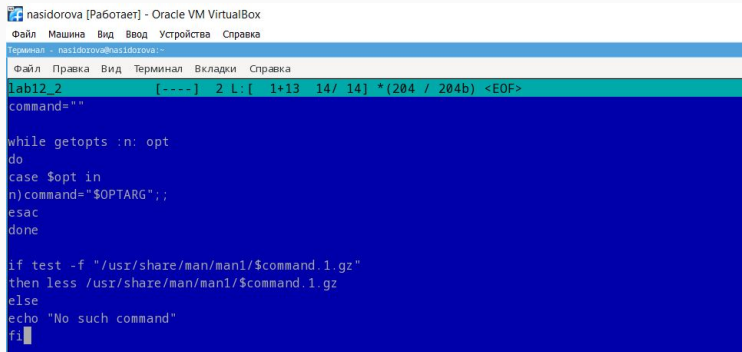
Рис. 1: Код к первому заданию

```
[nasidorova@nasidorova ~]$ mcedit lab12_1  
  
[nasidorova@nasidorova ~]$ chmod +x lab12_1  
[nasidorova@nasidorova ~]$ ./lab12_1  
file was locked  
unlocking  
file was locked  
unlocking  
file was locked  
unlocking  
file was locked
```

Рис. 2: Работа первого кода

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`.

Код задания 2



```
nasidorova [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Терминал - nasidorova@nasidorova:~
Файл  Правка  Вид  Терминал  Вкладки  Справка
lab12_2      [----]  2 L: [ 1+13  14/ 14] *(204 / 204b) <EOF>
command= ""

while getopts :n: opt
do
case $opt in
n)command="$OPTARG";;
esac
done

if test -f "/usr/share/man/man1/$command.1.gz"
then less /usr/share/man/man1/$command.1.gz
else
echo "No such command"
fi
```

Рис. 3: Код ко второму заданию

Работа кода 2

pasidoroja [Pasidoroja] - Oracle VM VirtualBox

Goła Macieja Bug Baci Wojciecha Czapiewa

openbsd - maildemon@localhost:

chciał: Pasidoroja Bug Baci Wojciecha Czapiewa

TOUCH(1)

User Commands

TOUCH(1)

```
#SC [tmMMH][S](M)
touch - change file timestamps

#SC [mSYNOPISS][S](M)
[SC] [mtouch][S](22m)[SC] [4mOPTION][S](24m)... [SC] [4mFILE][S](24m...)

#SC [mdDESCRIPTION][S](M)
Update the access and modification times of each FILE to the current time.

A FILE argument that does not exist is created empty, unless [SC] [im-c][S](22mox)[SC] [im-h][S](22mis supplied).

A FILE argument string of - is handled specially and causes touch to change the times of the file associated with standard output.

Mandatory arguments to long options are mandatory for short options too.

#SC [im-a][S](22mchange only the access time

#SC [im-C][S](22m, [SC] [im--no-create][S](M)(M)
do not create any files

#SC [im-[S](22m, [SC] [im--date][S](22m)[SC] [4mSTRING][S](M)
parse STRING and use it instead of current time

#SC [im-f][S](22m(ignore)

#SC [im-H][S](22m, [SC] [im--no-dereference][S](M)
affect each symbolic link instead of any referenced file (useful only on systems that can change the timestamps of a symlink)

#SC [im-m][S](22mchange only the modification time

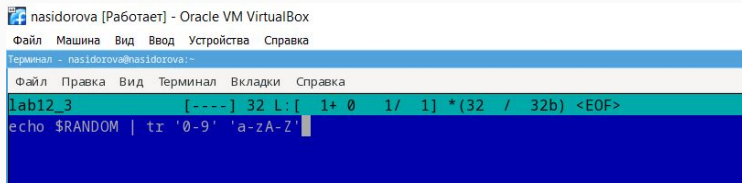
#SC [im-R][S](22m, [SC] [im--reference][S](22m)[SC] [4mFILE][S](M)
use this file's times instead of current time

#SC [im-t][S](22mSTAMP
use [[CCYY]MMDDhhmm[[ss]] instead of current time

/usr/share/man/man1/touch.1.gz
```

Рис. 4: Работа второго кода

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.



The screenshot shows a terminal window titled "nasidorova [Работает] - Oracle VM VirtualBox". The terminal has a menu bar with "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". Below the menu bar, the prompt is "Терминал - nasidorova@nasidorova:~". The terminal content shows a command being entered: "lab12_3 [----] 32 L:[1+ 0 1/ 1] *(32 / 32b) <EOF>". The command is "echo \$RANDOM | tr '0-9' 'a-zA-Z'".

```
nasidorova [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Терминал - nasidorova@nasidorova:~
Файл  Правка  Вид  Терминал  Вкладки  Справка
lab12_3 [----] 32 L:[ 1+ 0 1/ 1] *(32 / 32b) <EOF>
echo $RANDOM | tr '0-9' 'a-zA-Z'
```

Рис. 5: Код к третьему заданию

```
[nasidorova@nasidorova ~]$ mcedit lab12_3  
  
[nasidorova@nasidorova ~]$ chmod +x lab12_3  
[nasidorova@nasidorova ~]$ ./lab12_3  
bfife  
[nasidorova@nasidorova ~]$ ./lab12_3  
begbi  
[nasidorova@nasidorova ~]$ ./lab12_3  
ceaaa  
[nasidorova@nasidorova ~]$
```

Рис. 6: Работа третьего кода

В процессе выполнения этой лабораторной работы я продолжила осваивать программирование на `bash`.