

Отчет по лабораторной работе №12

Операционные системы

Лукина Р.О.

27 апреля 2023

Российский университет дружбы народов, Москва, Россия

Информация

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

- Лукина Ирина Олеговна
- студентка НБИ-04-22

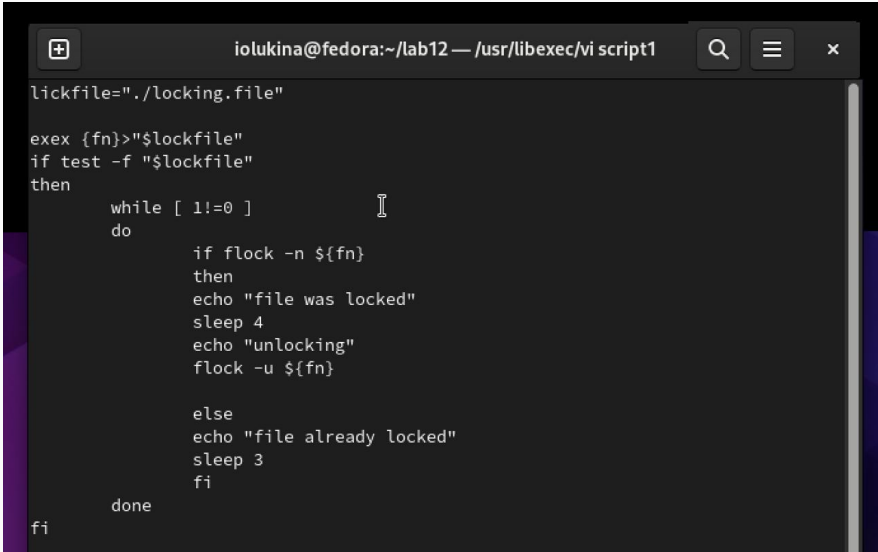
Вводная часть

- Освоить программирование внутри оболочки системы Линукс

Ход работы

Упрощенный механизм семафоров

- Создаю командный файл и проверяю его работу в терминале



The screenshot shows a terminal window with the title bar "iolukina@fedora:~/lab12 — /usr/libexec/vi script1". The terminal content is a shell script that implements a simplified semaphore mechanism using flock. The script defines a lockfile, sets up a function 'exex' to execute a command with a lock, and includes a while loop to wait for the lock to become available. The script is as follows:

```
lockfile="./locking.file"

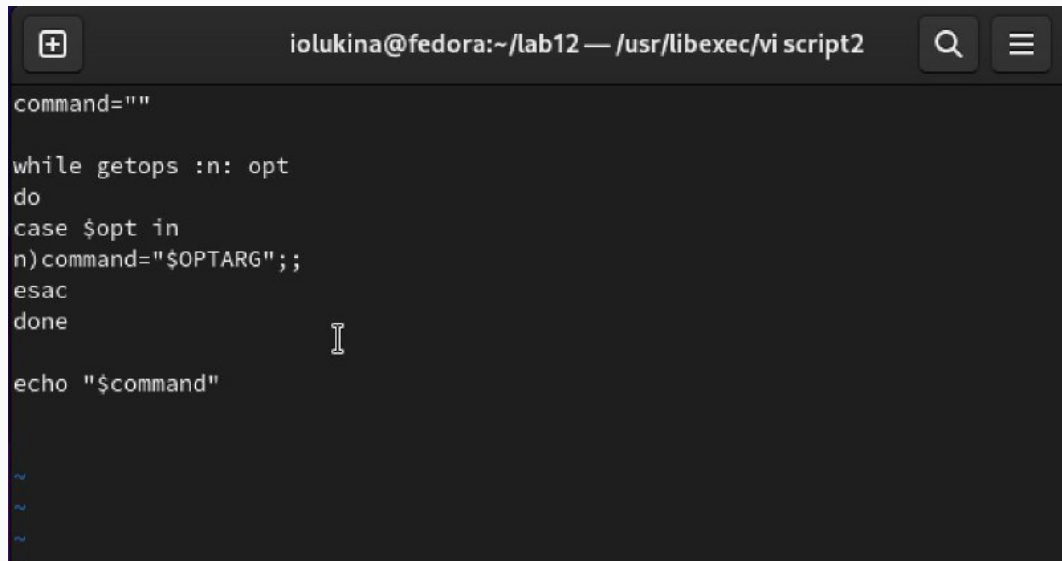
exex {fn}>"$lockfile"
if test -f "$lockfile"
then
    while [ 1!=0 ]
    do
        if flock -n ${fn}
        then
            echo "file was locked"
            sleep 4
            echo "unlocking"
            flock -u ${fn}

        else
            echo "file already locked"
            sleep 3
        fi
    done
fi
```

```
file was locked  
unlocking  
file was locked  
unlocking  
file was locked  
unlocking  
file was locked  
unlocking  
file was locked  
unlocking  
file was locked  
aunlocking  
file was locked  
aunlocking  
file was locked  
unlocking  
file was locked
```


- Проверяю каталог, создаю код и проверяю работу

```
[iolukina@fedora ~]$ cd /usr/share
[iolukina@fedora share]$ cd man
[iolukina@fedora man]$ cd man1
[iolukina@fedora man1]$ ls
:.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
```



The image shows a terminal window with a dark background. The title bar at the top contains a plus icon, the text 'iolukina@fedora:~/lab12 — /usr/libexec/vi script2', a search icon, and a menu icon. The terminal content shows a shell script with the following lines: 'command=""', 'while getopts :n: opt', 'do', 'case \$opt in', 'n) command="\$OPTARG";', 'esac', 'done', and 'echo "\$command"'. A cursor is positioned at the end of the 'done' line. On the left side of the terminal, there are three blue tilde characters '~' stacked vertically.

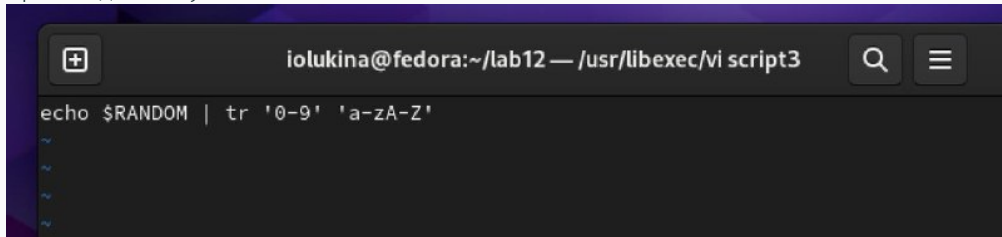
```
command=""

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

```
[iolukina@fedora lab12]$ vi script2  
[iolukina@fedora lab12]$ chmod 777 script2  
[iolukina@fedora lab12]$ ./script2 -n asd  
asd
```

Рис. 3: работа файла

- Пишу код, который будет генерировать случайную последовательность букв алфавита при каждом запуске

A terminal window with a dark background and a purple title bar. The title bar contains a plus icon on the left, the text 'iolukina@fedora:~/lab12 — /usr/libexec/vi script3' in the center, and search and menu icons on the right. The terminal shows the command 'echo \$RANDOM | tr '0-9' 'a-zA-Z'' followed by four tilde characters on separate lines.

```
iolukina@fedora:~/lab12 — /usr/libexec/vi script3
echo $RANDOM | tr '0-9' 'a-zA-Z'
~
~
~
~
```

```
ciafd  
[iolukina@fedora lab12]$ ./script3  
bbiaj  
[iolukina@fedora lab12]$ ./script3  
igfh  
[iolukina@fedora lab12]$ ./script3  
befab  
[iolukina@fedora lab12]$
```

Рис. 4: скрипт

- В ходе работы я приобрела практические навыки работы в системе Линукс