

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

**Факультет физико-математических и естественных
наук**

**Фундаментальная Информатика и Информационные
технологии**

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 12

дисциплина: операционные системы

Этук Нсе-Абаси Акпан

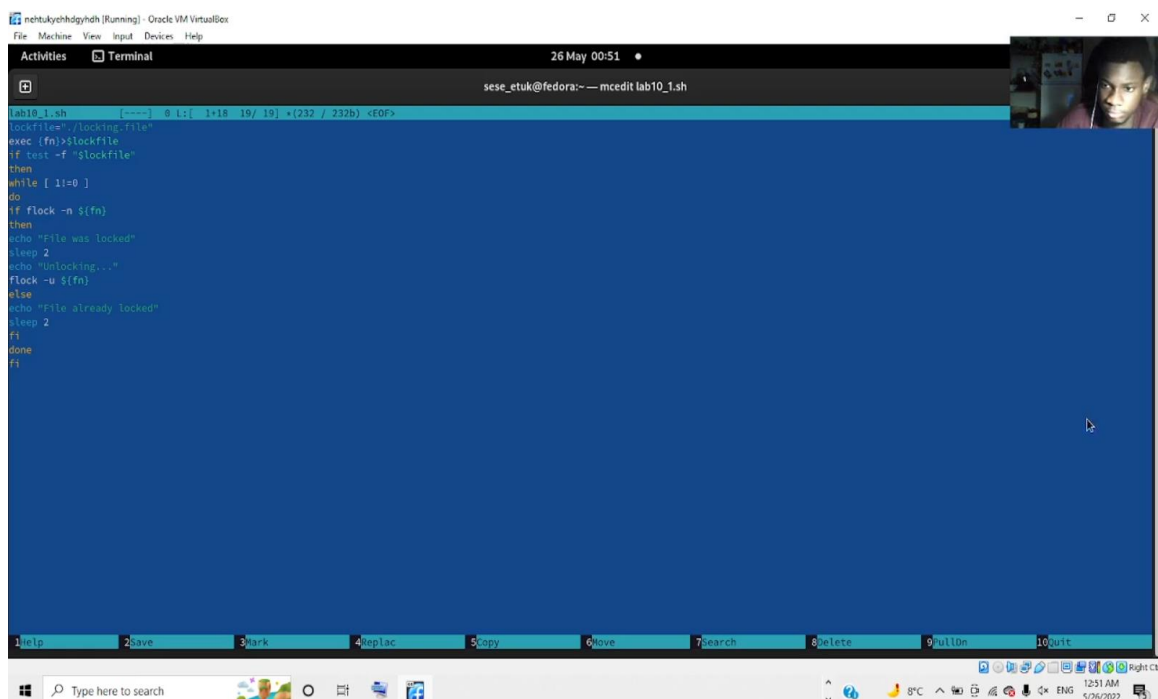
НФИбд-02-21

Цель работы

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

Ход работы

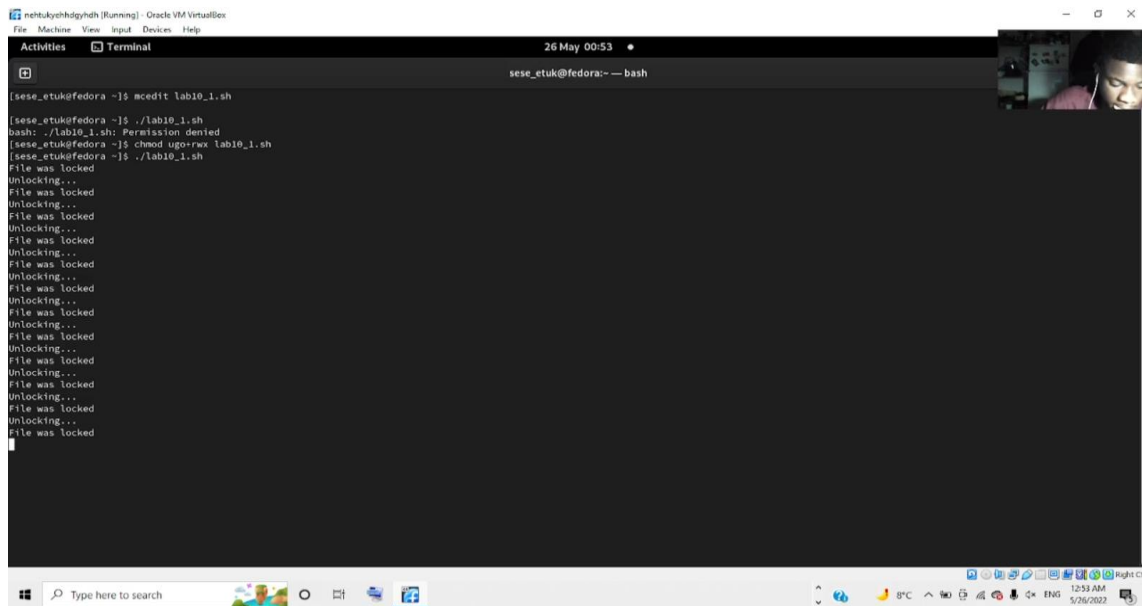
1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени t_1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени $t_2 < t_1$, также выдавая информацию 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.



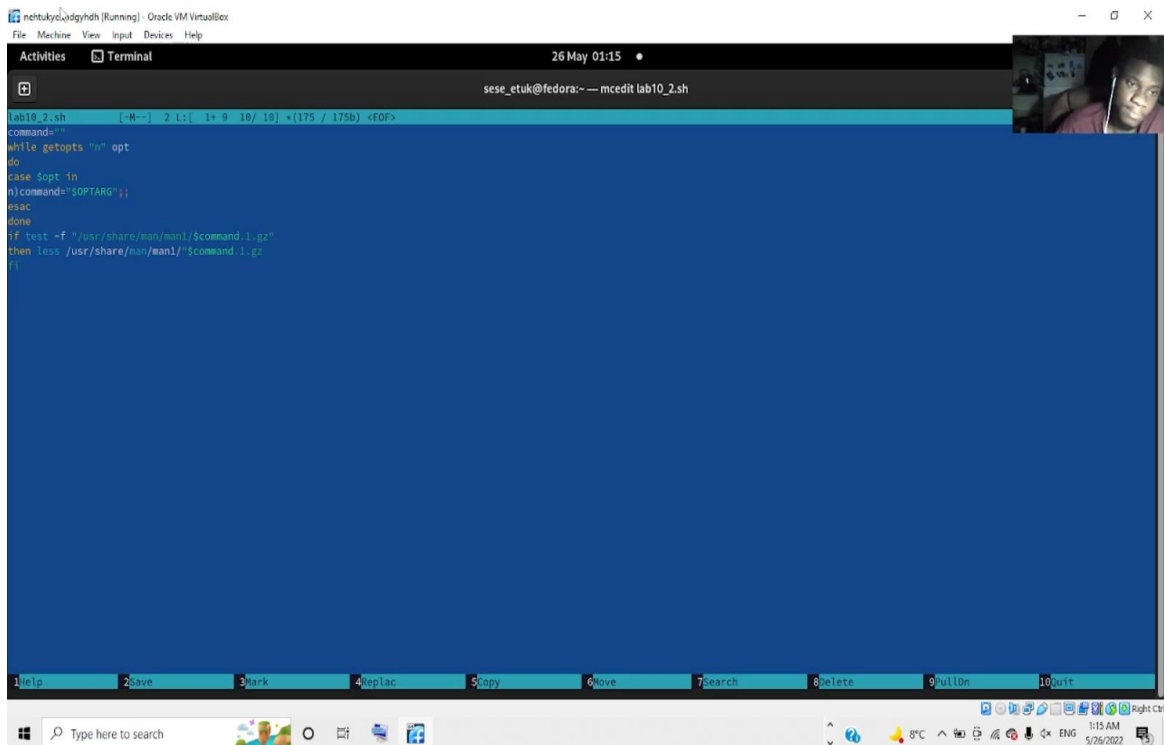
The screenshot shows a terminal window titled "nehtukychdyhdyh (Running) - Oracle VM VirtualBox". The terminal output is as follows:

```
lab10_1.sh [-----] 0 L: [ 1+18 19/ 19] * (232 / 232b) <EOF>
lockfiles"/locking_file"
exec (fn)>lockfile
if test -f 'lockfile'
then
while [ 1!=0 ]
do
if flock -n $(fn)
then
echo "File was locked"
sleep 2
echo "Unlocking..."
flock -u $(fn)
else
echo "File already locked"
sleep 2
fi
done
fi
```

The terminal window has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". The status bar at the bottom shows "26 May 00:51" and "sese_etuk@fedora:~ -- mcedit lab10_1.sh". The terminal window is running on a Windows 10 desktop environment.



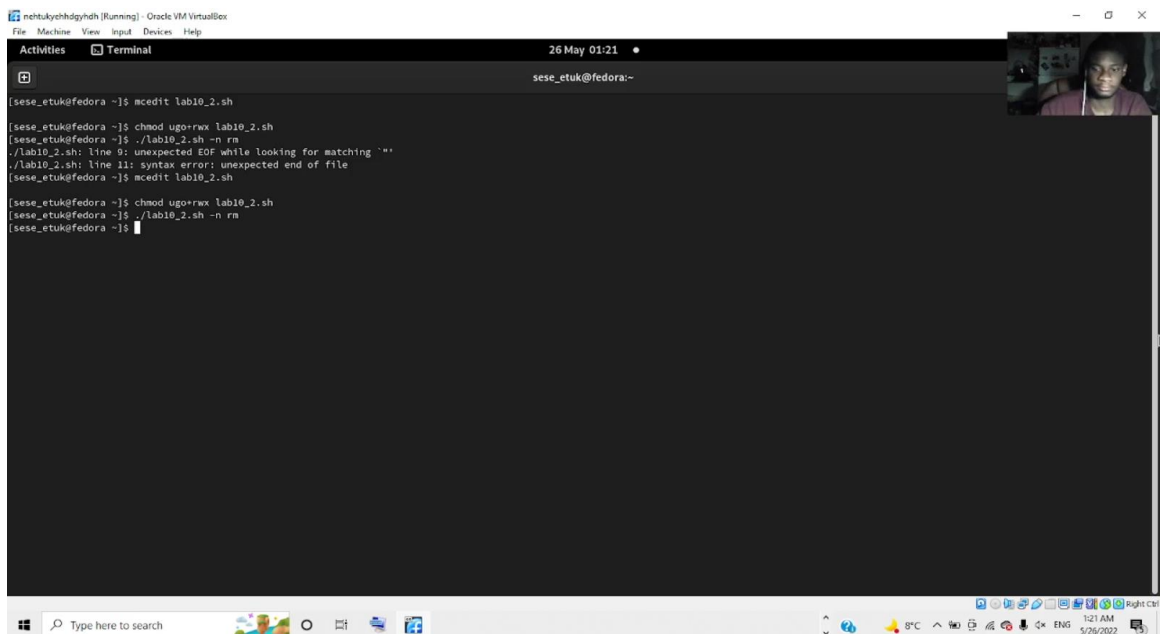
2. Реализовать команду `man` с помощью командного файла. Изучите содержимое каталога `/usr/share/man/man1`. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой `less` сразу же просмотрев содержимое справки. 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`.



```
nehtukycldgyhdi (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
26 May 01:15
sese_etuk@fedora:~ -- mcedit lab10_2.sh

lab10_2.sh [-M-] 2 L: [ 1+ 9 10/ 10] * (175 / 175b) <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
fi

1:help 2:save 3:mark 4:replac 5:copy 6:move 7:search 8:delete 9:undo 10:quit
Type here to search 1:15 AM 5/26/2022
```



```
nehtukycldgyhdi (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
26 May 01:21
sese_etuk@fedora:~

[sese_etuk@fedora ~]$ mcedit lab10_2.sh

[sese_etuk@fedora ~]$ chmod ugo+rx lab10_2.sh
[sese_etuk@fedora ~]$ ./lab10_2.sh -n rm
./lab10_2.sh: line 0: unexpected EOF while looking for matching `''
./lab10_2.sh: line 11: syntax error: unexpected end of file
[sese_etuk@fedora ~]$ mcedit lab10_2.sh

[sese_etuk@fedora ~]$ chmod ugo+rx lab10_2.sh
[sese_etuk@fedora ~]$ ./lab10_2.sh -n rm
[sese_etuk@fedora ~]$
```

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.

```
nehtukyhgdghdh (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help
26 May 01:41
sese_etuk@fedora:~ -- mcedit lab10_3.sh

lab10_3.sh [sese] 0 1: 1* 6 7/ 7) *(167 / 167b) <EOF>
[sese_etuk@fedora:~] cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 10 | head -n 1
[sese_etuk@fedora:~] echo "Prog by Your yech requirement: "
[sese_etuk@fedora:~] echo $RANDOM | tr '0-9' 'a-zA-Z'
```

```
nehtukyhgdghdh (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help
26 May 01:44
sese_etuk@fedora:~

[sese_etuk@fedora:~] mcedit lab10_2.sh
[sese_etuk@fedora:~] chmod ugo+rxw lab10_2.sh
[sese_etuk@fedora:~] ./lab10_2.sh -n rm
./lab10_2.sh: line 9: unexpected EOF while looking for matching `"'
./lab10_2.sh: line 11: syntax error: unexpected end of file
[sese_etuk@fedora:~] mcedit lab10_2.sh
[sese_etuk@fedora:~] chmod ugo+rxw lab10_2.sh
[sese_etuk@fedora:~] ./lab10_2.sh -n rm
./lab10_2.sh: Permission denied
[sese_etuk@fedora:~] chmod ugo+rxw lab10_3.sh
[sese_etuk@fedora:~] mcedit lab10_3.sh
[sese_etuk@fedora:~] ./lab10_3.sh -n rm
bash: ./lab10_3.sh: Permission denied
[sese_etuk@fedora:~] chmod ugo+rxw lab10_3.sh
[sese_etuk@fedora:~] ./lab10_3.sh
[sese_etuk@fedora:~] My password generator
AgCJtkxXoe
[sese_etuk@fedora:~] Prog by Your yech requirement:
bbgtid
[sese_etuk@fedora:~] ./lab10_3.sh
[sese_etuk@fedora:~] My password generator
SQF0FEybo0
[sese_etuk@fedora:~] Prog by Your yech requirement:
cdeda
[sese_etuk@fedora:~] mcedit lab10_3.sh
[sese_etuk@fedora:~] chmod ugo+rxw lab10_3.sh
[sese_etuk@fedora:~] ./lab10_3.sh
[sese_etuk@fedora:~] My password generator
qfKocwIFw
[sese_etuk@fedora:~] Prog by Your tech requirement:
dcba1
[sese_etuk@fedora:~] ./lab10_3.sh
[sese_etuk@fedora:~] My password generator
nrwX1B1tq0
[sese_etuk@fedora:~] Prog by Your tech requirement:
dcffj
[sese_etuk@fedora:~]
```

Вывод

Анализ результатов

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

Контрольные вопросы

- 1: Найдите синтаксическую ошибку в следующей строке: `while [$1 != "exit"] $1`. Так же между скобками должны быть пробелы. В противном случае скобки и рядом стоящие символы будут восприниматься как одно целое
- 2: Как объединить (конкатенация) несколько строк в одну? `cat file.txt | xargs | sed -e 's/. /\n/g'`
- 3: Найдите информацию об утилите `seq`. Какими иными способами можно реализовать её функционал при программировании на `bash`? `seq` - выдает последовательность чисел. Реализовать ее функционал можно командой `for n in {1..5} do <КОМАНДА> done`
- 4: Какой результат даст вычисление выражения `$((10/3))`? 3
- 5: Укажите кратко основные отличия командной оболочки `zsh` от `bash`. `Zsh` очень сильно упрощает работу. Но существуют различия. Например, `zsh` после `for` обязательно вставляет пробел, нумерация массивов `zsh` начинается с 1 (что не особо удобно на самом деле). Если вы собираетесь писать скрипт, который легко будет запускать множество разработчиков, то я рекомендую `Bash`. Если скрипты вам не нужны - `Zsh` (более простая работа с файлами, например)
- 6: Проверьте, верен ли синтаксис данной конструкции `for ((a=1; a <= LIMIT; a++))` Верен
- 7: Сравните язык `bash` с какими-либо языками программирования. Какие преимущества у `bash` по сравнению с ними? Какие недостатки? `Bash` позволяет очень легко работать с файловой системой без лишней конструкции (в отличие от обычного языка программирования). Но относительно обычных языков программирования `bash` очень сжат. Тот же `Си` имеет гораздо более широкие возможности для разработчика.