

Лабораторная работа No 13

**Программирование в командном процессоре ОС UNIX. Расширенное
программирование**

Кеан Путхеаро НПИбд-01-20

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Вывод	10
5	Библиография	11

Список иллюстраций

3.1	vi	6
3.2	ВЫВОД	6
3.3	vi	7
3.4	vi	7
3.5	ВЫВОД	8
3.6	mcredit	8
3.7	ВЫВОД	9

1 Цель работы

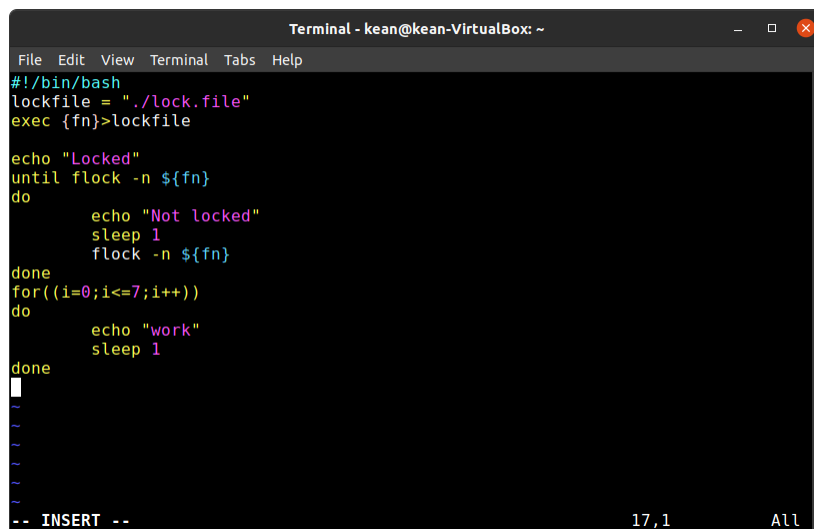
Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

2 Задание

1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени 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.
2. Реализовать команду `man` с помощью командного файла. Изучите содержимое каталога `/usr/share/man/man1`. В нём 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`.
3. Используя встроенную переменную `RANDOM`, — `., RANDOM` выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

3 Выполнение лабораторной работы

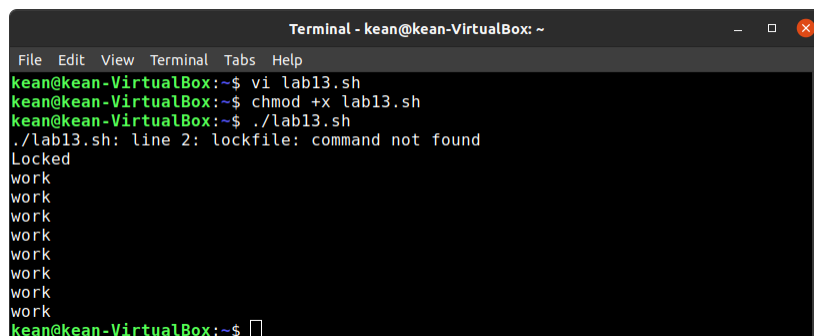
1. Написал командный файл, реализующий упрощённый механизм семафоров.



```
Terminal - kean@kean-VirtualBox: ~
File Edit View Terminal Tabs Help
#!/bin/bash
lockfile = "./lock.file"
exec {fn}>lockfile

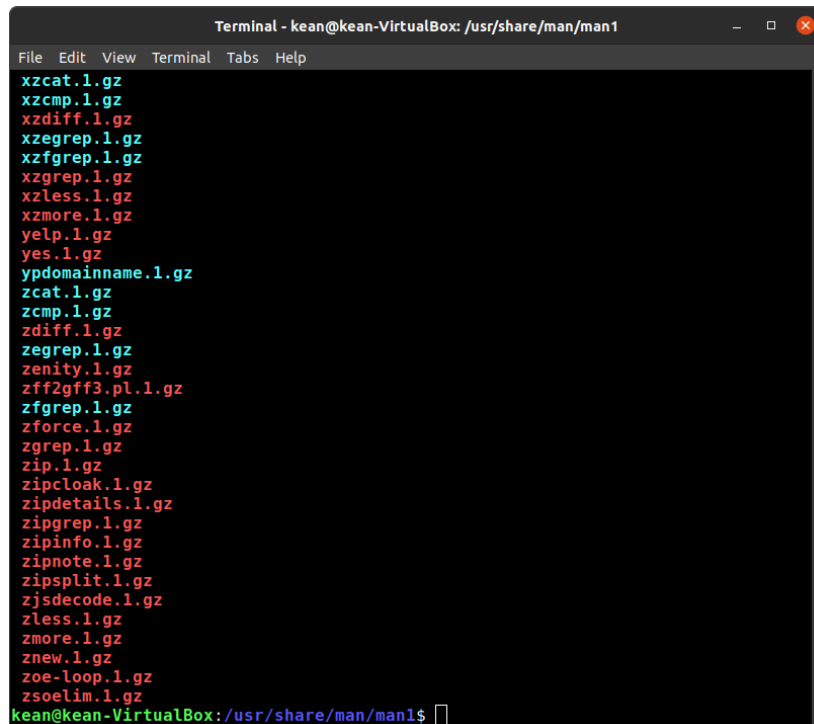
echo "Locked"
until flock -n ${fn}
do
    echo "Not locked"
    sleep 1
    flock -n ${fn}
done
for((i=0;i<=7;i++))
do
    echo "work"
    sleep 1
done
-- INSERT --
```

Рис. 3.1: vi

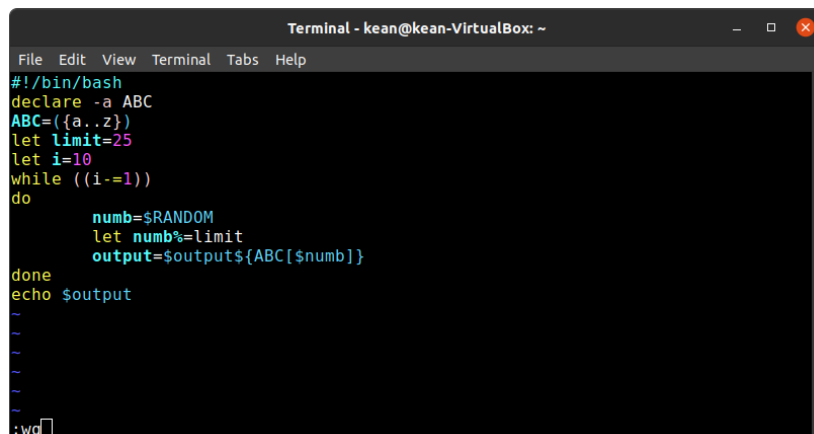


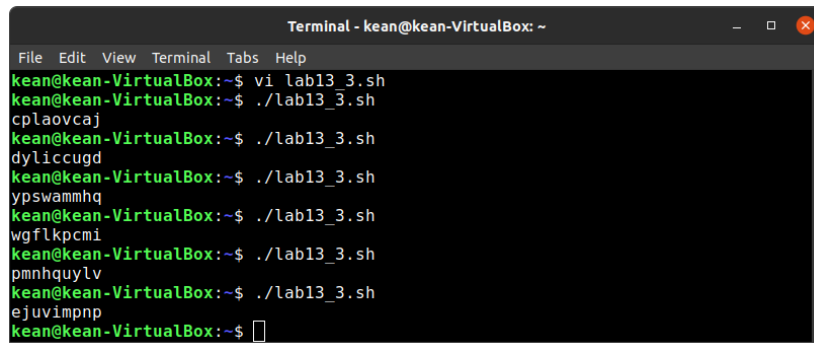
```
Terminal - kean@kean-VirtualBox: ~
File Edit View Terminal Tabs Help
kean@kean-VirtualBox:~$ vi lab13.sh
kean@kean-VirtualBox:~$ chmod +x lab13.sh
kean@kean-VirtualBox:~$ ./lab13.sh
./lab13.sh: line 2: lockfile: command not found
Locked
work
work
work
work
work
work
work
work
work
kean@kean-VirtualBox:~$
```

Рис. 3.2: вывод



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.



A terminal window titled "Terminal - kean@kean-VirtualBox: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows a series of commands and outputs. The user enters "vi lab13_3.sh", then ". /lab13_3.sh". This is followed by a loop of commands and outputs: "cplaovcaj", "dyliccugd", "ypswammhq", "wgflkpcmi", "pmnhquylv", and "ejuvimpnp". Each command is followed by a prompt "kean@kean-VirtualBox:~\$" and the command ". /lab13_3.sh". The terminal ends with a prompt "kean@kean-VirtualBox:~\$" and a cursor.

```
Terminal - kean@kean-VirtualBox: ~
File Edit View Terminal Tabs Help
kean@kean-VirtualBox:~$ vi lab13_3.sh
kean@kean-VirtualBox:~$ ./lab13_3.sh
cplaovcaj
kean@kean-VirtualBox:~$ ./lab13_3.sh
dyliccugd
kean@kean-VirtualBox:~$ ./lab13_3.sh
ypswammhq
kean@kean-VirtualBox:~$ ./lab13_3.sh
wgflkpcmi
kean@kean-VirtualBox:~$ ./lab13_3.sh
pmnhquylv
kean@kean-VirtualBox:~$ ./lab13_3.sh
ejuvimpnp
kean@kean-VirtualBox:~$
```

Рис. 3.7: вывод

4 Вывод

В результате работы, я изучил основы программирования в оболочке ОС UNIX. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов

5 Библиография

1. (Лабораторная работа №12) https://esystem.rudn.ru/pluginfile.php/1142380/mod_resource/content/1/lab_shell_prog_2.pdf
2. (stackexchange) <https://vi.stackexchange.com/questions/10209/execute-current-buffer-as-bash-script-from-vim>
3. (BASH: функция getopt — используем опции в скриптах) https://esystem.rudn.ru/pluginfile.php/1142380/mod_resource/content/1/lab_shell_prog_2.pdf
4. (stackoverflow) <https://stackoverflow.com/questions/16483119/an-example-of-how-to-use-getopts-in-bash>