

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Кеан Путхеаро НПИбд-01-20¹

28 мая, 2021, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

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

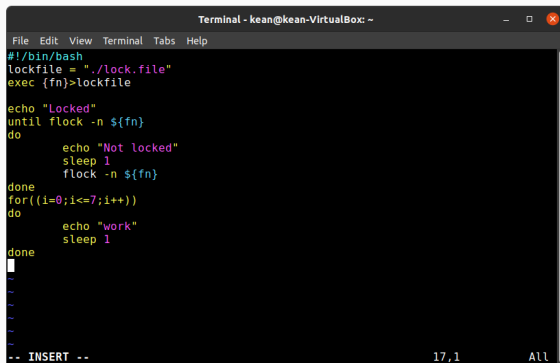
Задачи лабораторной работы

1. Написать командный файл, реализующий упрощённый механизм семафоров. Ко-мандный 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 (про-цессом). 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 в фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов.

Задачи лабораторной работы

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 ко-манд. 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.

Процесс выполнения лабораторной работы

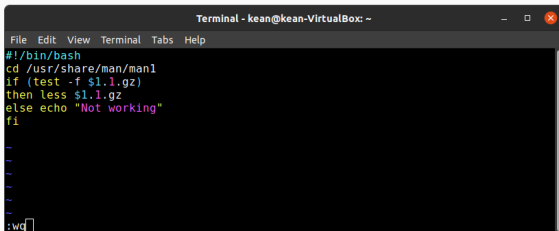


The image shows a terminal window titled "Terminal - kean@kean-VirtualBox: ~". The window contains a bash script that demonstrates the use of the `flock` command for file locking. The script sets a lockfile to `./lock.file` and uses `exec {fn}>lockfile` to open it. It then enters a loop where it checks if the lock is held by another process. If not, it prints "Not locked", sleeps for 1 second, and then locks the file. The script also includes a `for` loop that prints "work" and sleeps for 1 second. The terminal output shows the script running successfully, with the lock being acquired and released as expected. The bottom of the terminal shows the `-- INSERT --` prompt and the cursor position at line 17, column 1.

```
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 -- 17,1 All
```

Рис. 1: vi

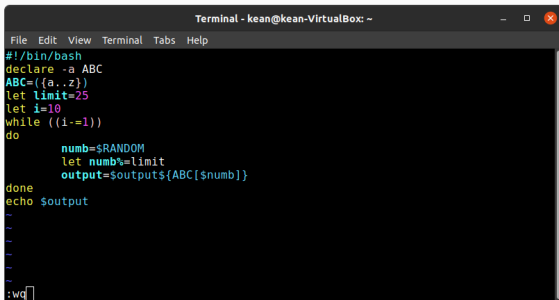


The image shows a terminal window titled "Terminal - kean@kean-VirtualBox: ~". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal content shows a shell script being edited with the vi editor. The script is as follows:

```
#!/bin/bash
cd /usr/share/man/man1
if (test -f $1.1.gz)
then less $1.1.gz
else echo "Not working"
fi
```

The prompt is `:wq` followed by a cursor, indicating the user is in write mode and ready to save the file.

Рис. 2: vi



The image shows a terminal window titled "Terminal - kean@kean-VirtualBox: ~". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal content shows a shell script being edited with the vi editor. The script starts with a shebang line, followed by variable declarations and a while loop. The prompt ":wq" is visible at the bottom, indicating the user is saving and exiting the editor.

```
#!/bin/bash
declare -a ABC
ABC=({a..z})
let limit=25
let i=10
while ((i-=1))
do
    numb=$RANDOM
    let numb%=limit
    output=$output${ABC[numb]}
done
echo $output
~
~
~
~
:wq
```

Рис. 3: vi

Выводы по проделанной работе

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

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>