

Лабораторная работа 12

Программирование в командном процессоре ОС UNIX. Расширенное
программирование

Бешкуров Тимофей Борисович

Содержание

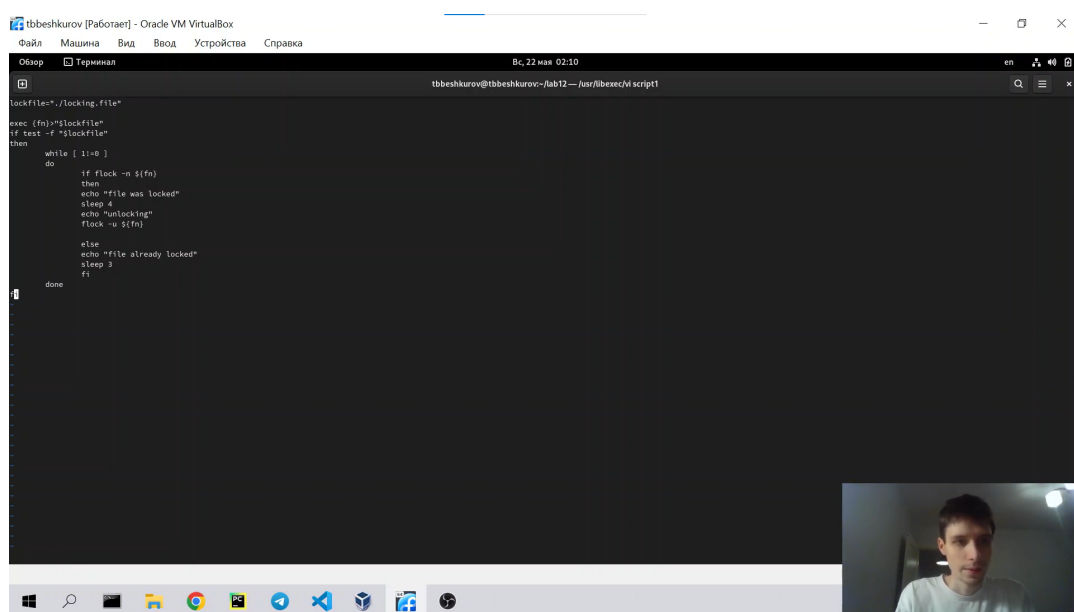
Цель работы	3
Выполнение лабораторной работы	4
Вывод	8
Контрольные вопросы	9
Контрольные вопросы	9

Цель работы

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

Выполнение лабораторной работы

1. Скрипт 1 (рис. 1)



```
lockfile="/locking_file"

exec {fn}>"$lockfile"
if test -f "$lockfile"
then
    while [ 1=1 ]
    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
```

Рис. 1: Скрипт 1

Запуск (рис. 2)

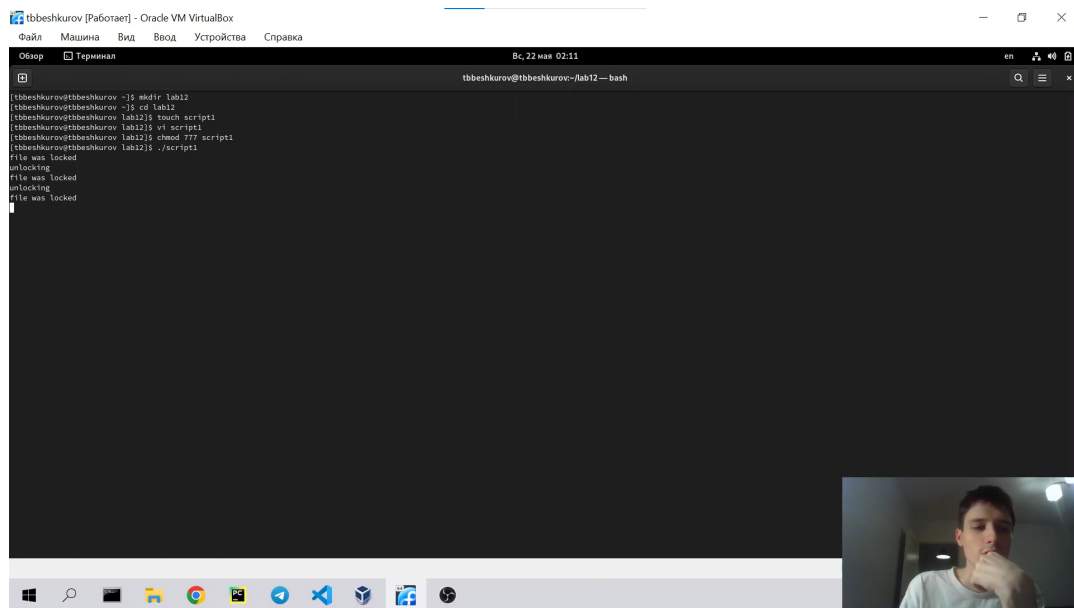


Рис. 2: Скрипт 1 запуск

2. Скрипт 2 (рис. 3)

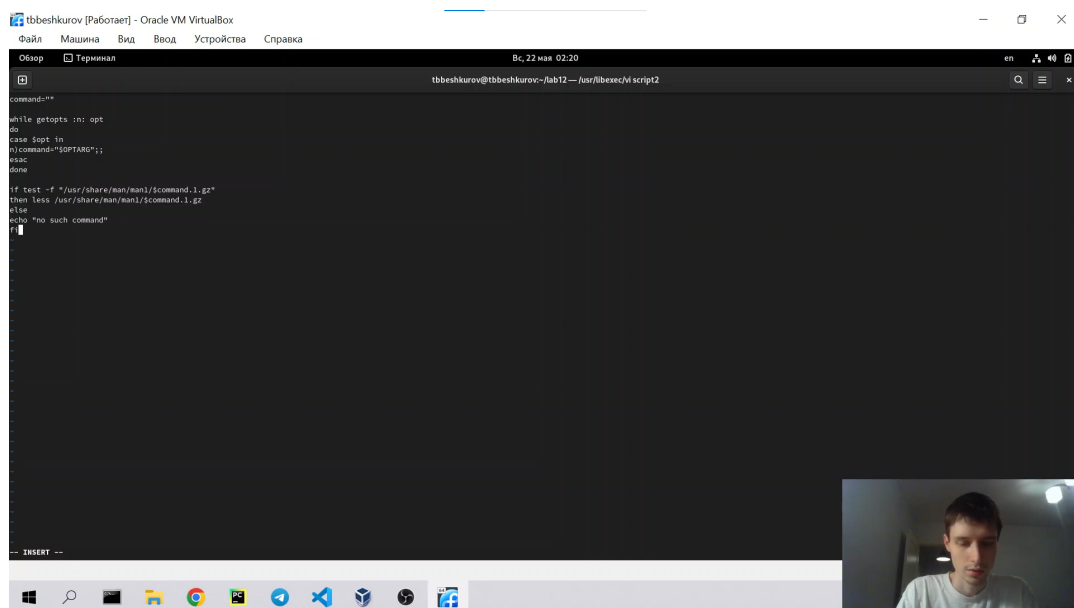


Рис. 3: Скрипт 2

Запуск (рис. 4)

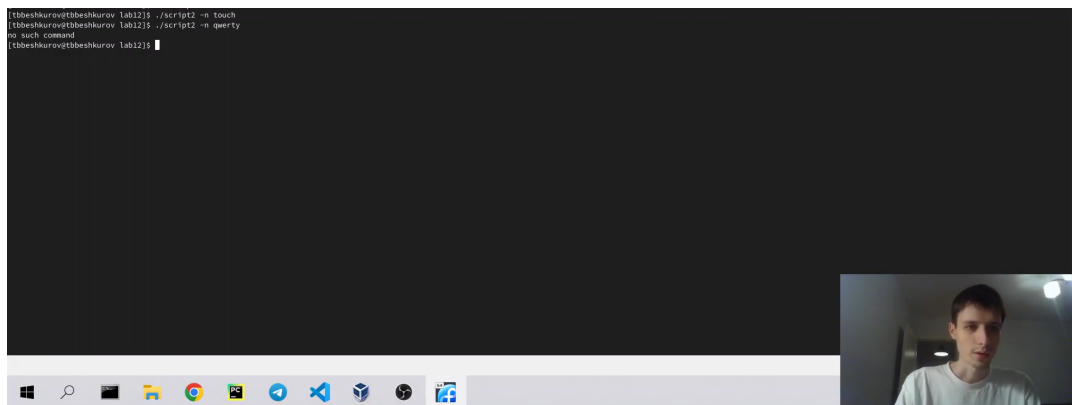


Рис. 4: Скрипт 2 запуск

3. Скрипт 3 (рис. 5)

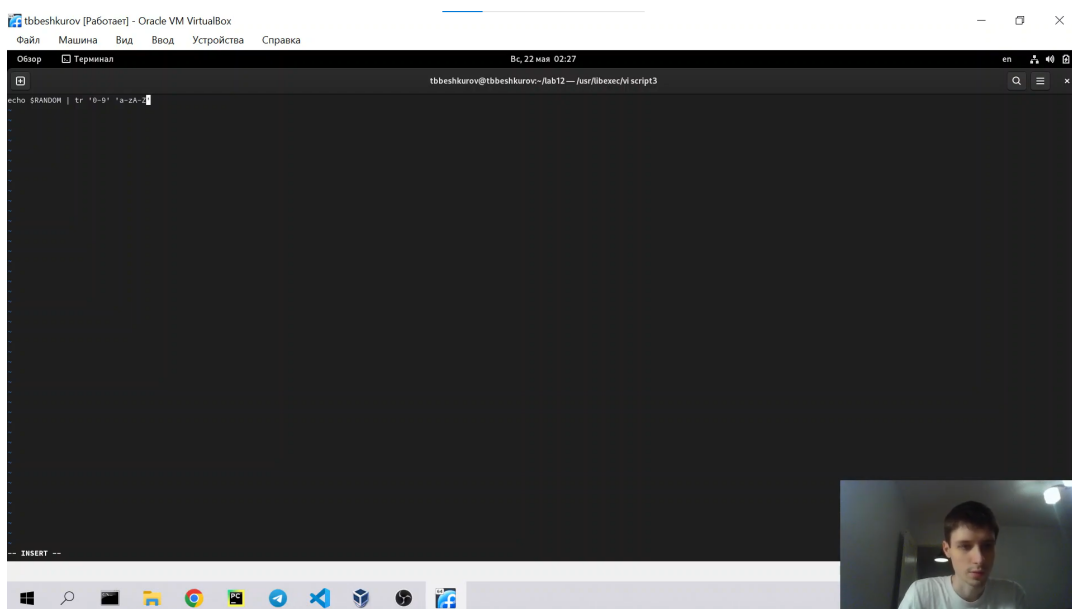


Рис. 5: Скрипт 3

Запуск (рис. 6)

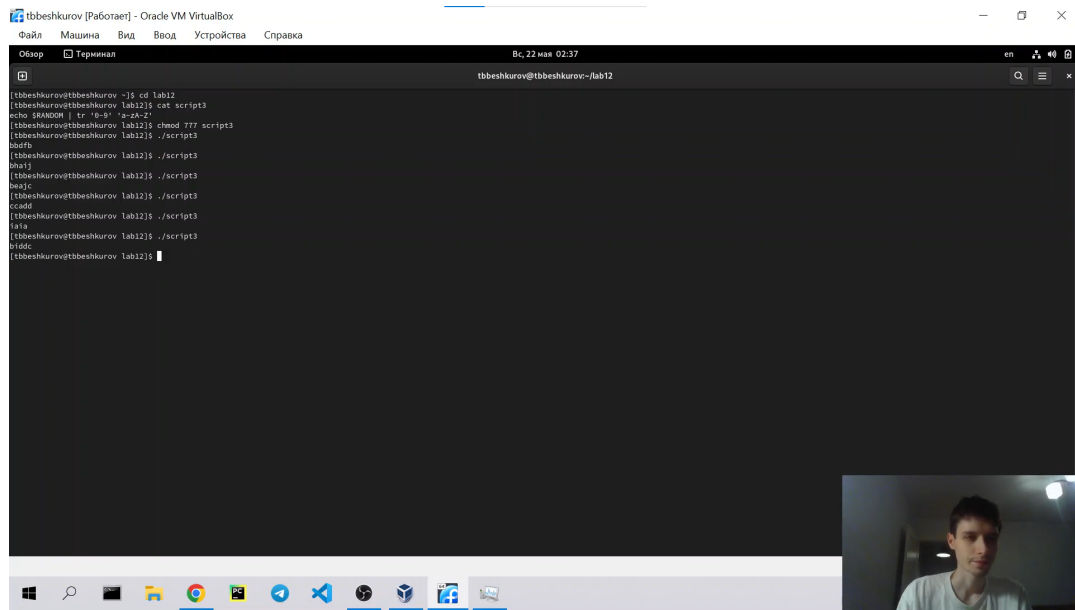


Рис. 6: Скрипт 3 запуск

Вывод

Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Контрольные вопросы

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 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, в `zsh` после `for` обязательно вставлять пробел, нумерация массивов в `zsh` начинается с 1 (что не особо удобно на самом деле). Если вы собираетесь писать скрипт, который легко будет запускать множество разработчиков, то я рекомендуется `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 лишних конструкций (в отличие от обычного языка программирования). Но относительно обычных языков программирования `bash` очень сжат. Тот же `C` имеет гораздо более широкие возможности для разработчика.