

Лабораторная работа №12

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

Перевощиков Данил Алексеевич

Содержание

1	Цель работы	5
2	Ход работы	6
3	Вывод	10
4	Контрольные вопросы	11

Список иллюстраций

2.1	Скрипт 1.	6
2.2	Выполнение скрипта 1.	7
2.3	Скрипт 2.	7
2.4	Выполнение скрипта 2.	8
2.5	Скрипт 3.	8
2.6	Выполнение скрипта 3.	9

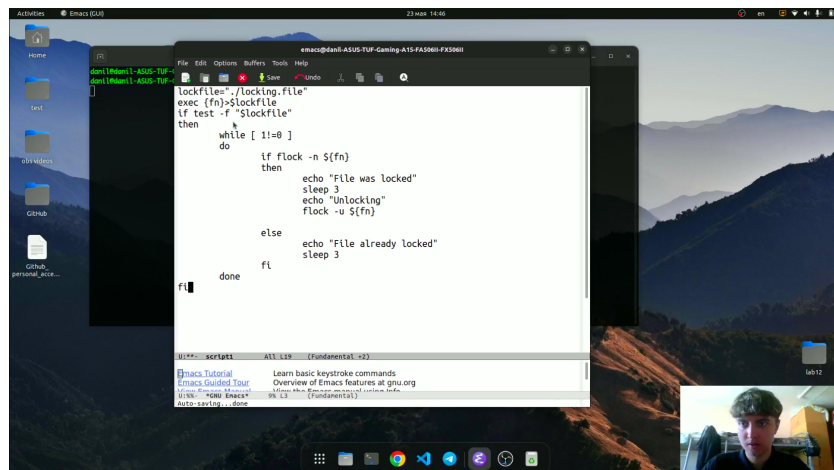
Список таблиц

1 Цель работы

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

2 Ход работы

1. Написали командный файл, реализующий упрощённый механизм семафоров. Командный файл в течение некоторого времени t_1 дожидается освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, используем его в течение некоторого времени $t_2 < t_1$, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). (рис. 2.1, 2.2)



```
lockfile="/locking.file"
exec (fn)>lockfile
if test -f "$lockfile"
then
  while [ 1!=0 ]
  do
    if flock -n $(fn)
    then
      echo "File was locked"
      sleep 3
      echo "Unlocking"
      flock -u $(fn)
    else
      echo "File already locked"
      sleep 3
    fi
  done
fi
```

Рис. 2.1: Скрипт 1.


```
danil@danil-ASUS-TUF-Gaming-A15-FA506II-FX506II: ~/Desktop/lab12
danil@danil-ASUS-TUF-Gaming-A15-FA506II-FX506II: ~/Desktop/lab12$ emacs script3
danil@danil-ASUS-TUF-Gaming-A15-FA506II-FX506II: ~/Desktop/lab12$ chmod +x script3
danil@danil-ASUS-TUF-Gaming-A15-FA506II-FX506II: ~/Desktop/lab12$ ./script3
password:
bmT8u1ueoRBNfh
danil@danil-ASUS-TUF-Gaming-A15-FA506II-FX506II: ~/Desktop/lab12$ ./script3
password:
YtqUUIVmx5Mq8Ke
danil@danil-ASUS-TUF-Gaming-A15-FA506II-FX506II: ~/Desktop/lab12$ ./script3
password:
ZP7CmyIECuP4eS
danil@danil-ASUS-TUF-Gaming-A15-FA506II-FX506II: ~/Desktop/lab12$ ./script3
password:
dGlt33dKxpgBwKY
danil@danil-ASUS-TUF-Gaming-A15-FA506II-FX506II: ~/Desktop/lab12$
```

Рис. 2.6: Выполнение скрипта 3.

3 Вывод

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

4 Контрольные вопросы

1. Найдите синтаксическую ошибку в следующей строке: `1 while [$1 != "exit"]`

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))`?

5.

6. 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 Си имеет гораздо более широкие возможности для разработчика.