Отчёта по лабораторной работе №13

Операционый Систем

Коне Сирики НФИБД-01-20

Содержание

List of Tables

List of Figures

1 Цель работы

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

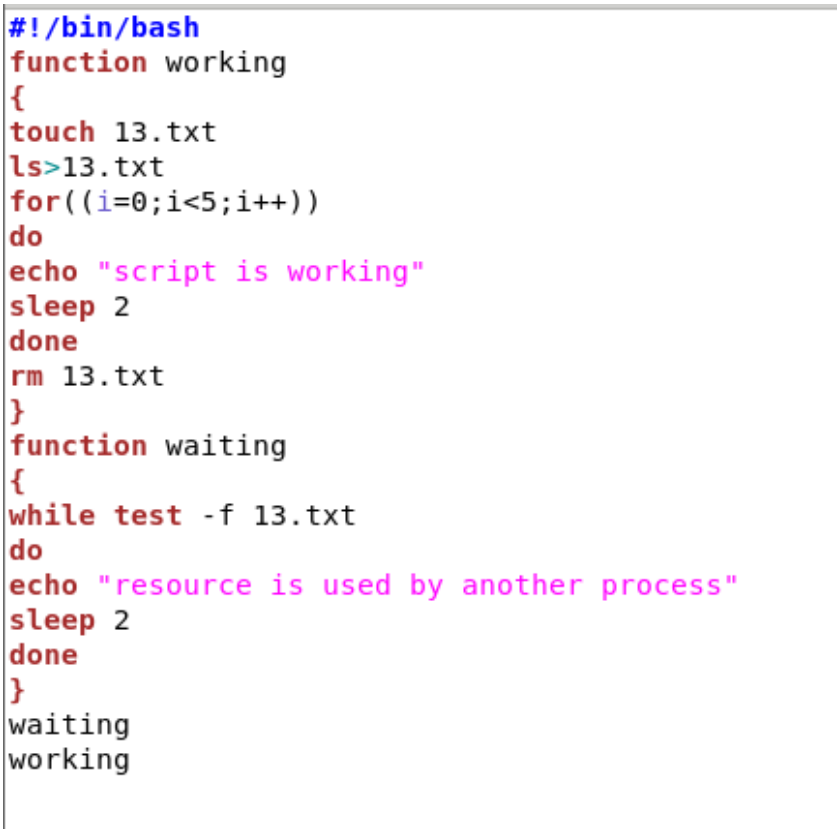
2 Задание

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

3 Выполнение лабораторной работы

Ход работы: 1. . Написал командный файл, реализующий упрощённый механизм семафоров. Командный файл в течение 2 сек. дожидается освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использует его в течение 10 сек., также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустила командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой, в котором также был запущен этот файл, но не фоновом, а в привилегированном режиме. Код скрипта (семафором в данном случае выступает файл 12.txt, т.е., по сути, запись в него (ls>12.txt бессмысленна.) При реальном использовании можно производить её в некоторый другой файл, т.е. именно этот другой файл будет занятым ресурсом, а 12.txt чисто «семафором»):

(рис. 3.1)



```
#!/bin/bash
function working
{
touch 13.txt
ls>13.txt
for((i=0;i<5;i++))
do
echo "script is working"
sleep 2
done
rm 13.txt
}
function waiting
{
while test -f 13.txt
do
echo "resource is used by another process"
sleep 2
done
}
waiting
working
```

Figure 3.1: рисунок 1

Его работа (1 скрин – второй терминал, сначала запускается скрипт и выводится первое сообщение о его работе, затем (на 2 скрине) из первого терминала запущен скрипт, он выводит сообщения о занятости ресурса во второй терминал, т.е., как видно, сообщения в итоге чередуются. А затем скрипт, запущенный в первом терминале переход из стадии ожидания в стадию выполнения и подряд выводятся несколько сообщений о его работе):

(рис. 3.2)

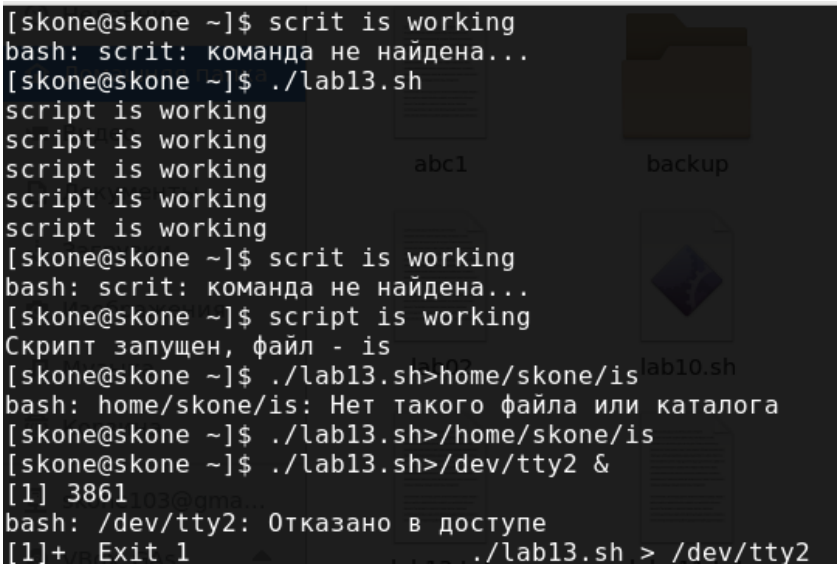


Figure 3.2: рисунок 2

2. Реализовал команду man с помощью командного файла. Изучила 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 получает в виде аргумента командной строки название команды и в виде результата выдаёт справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге man1. Код скрипта:

(рис. 3.3)

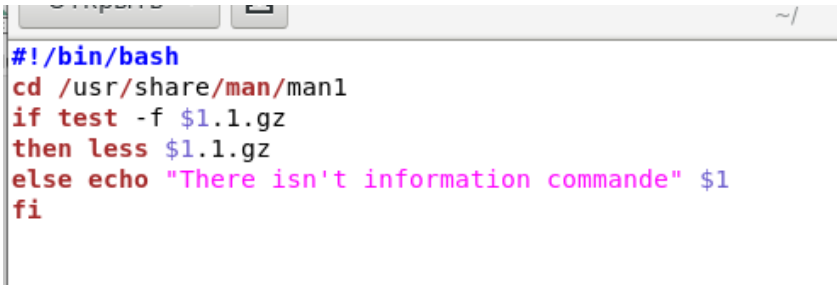


Figure 3.3: рисунок 3

(рис. 3.4)

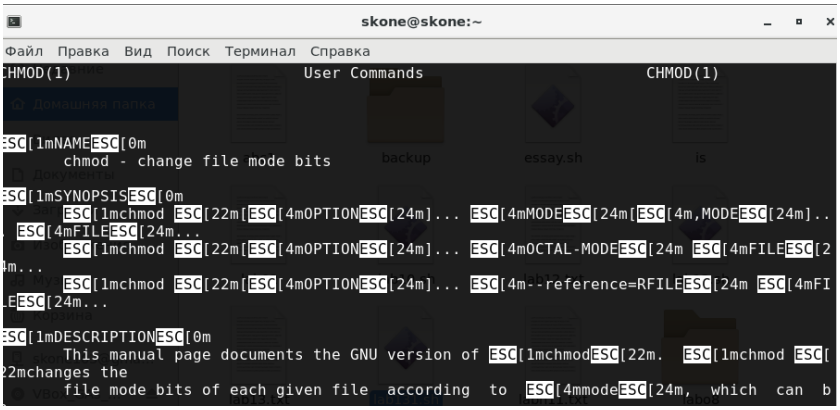


Figure 3.4: рисунок 4

(рис. 3.5)

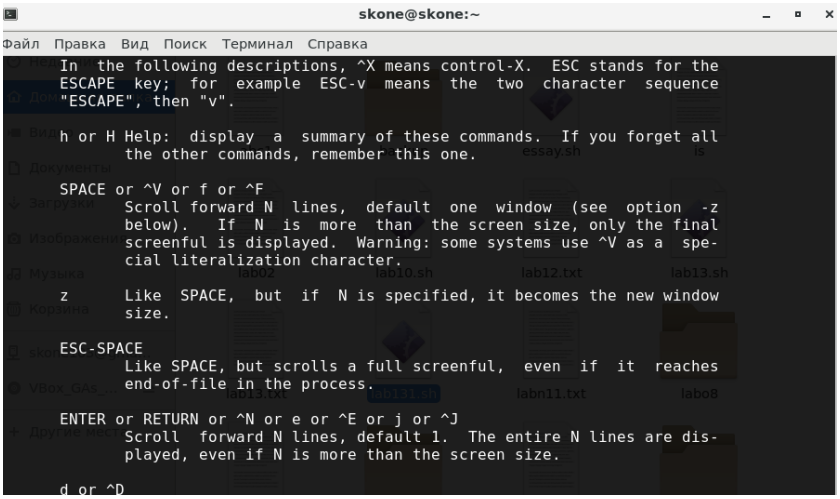


Figure 3.5: рисунок 5

3. Используя встроенную переменную $RANDOM, написала командный файл, генерирующий случайную последовательность букв латинского алфавита. Учла, что $RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767 (чтоб получился рандом от 1 до 52, берём остаток от деления $RANDOM на 52 и прибавляем у нему 1). Код скрипта (в файле 11.txt записаны все заглавные и строчные буквы латинского алфавита по одной в строке):

(рис. 3.6)

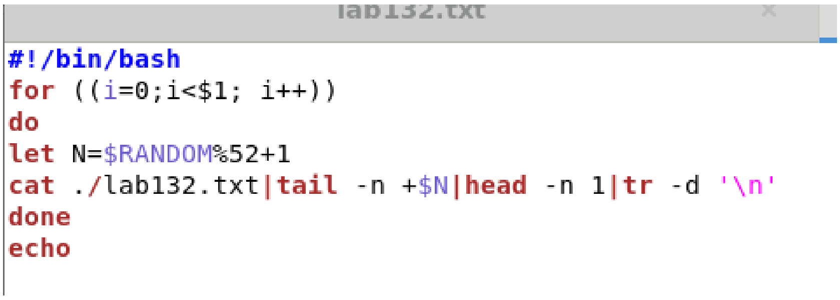


Figure 3.6: рисунок 6

(рис. 3.7)

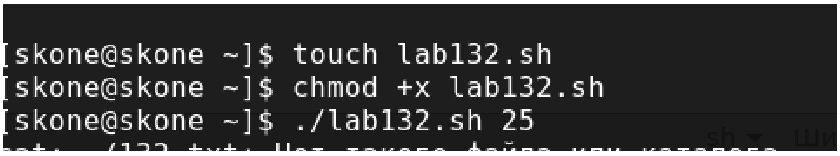


Figure 3.7: рисунок 7

(рис. 3.8)

Figure 3.8: рисунок 8

Контрольные вопросы:

1. Найдите синтаксическую ошибку в следующей строке: while [$1 != “exit”] Нужны пробелы после и перед открывающей и закрывающей скобками соответственно. Также желательно заключить $1 в кавычки (“$1”) во избежание ошибки, если $1 пуст.
2. Как объединить (конкатенация) несколько строк в одну? Str0=”str1str2”
3. Найдите информацию об утилите seq. Какими иными способами можно реализовать её функционал при программировании на bash? Seq выводит

последовательность целых чисел с шагом, заданным пользователем. Другая утилита с той же функцией – jot.

4. Какой результат даст вычисление выражения $((10/3))? Результатом будет 3.
5. Укажите кратко основные отличия командной оболочки zsh от bash. В zsh можно настраивать горячие клавиши. Автодополнение более сложное и гибкое. Используется большое количество различных опций, а также максимально краткий синтаксис. В итоге, zsh удобен для повседневной, рутинной работы, а для написания скриптов всё же лучше использовать bash.
6. Проверьте, верен ли синтаксис данной конструкции for ((a=1; a <= LIMIT; a++)) Синтаксис верен.
7. Сравните язык bash с какими-либо языками программирования. Какие преимущества у bash по сравнению с ними? Какие недостатки? Bash, как мне кажется, имеет достаточно много сходств с pythonом и с Си. Недостатком баша является достаточно нагруженный синтаксис (легко допустить ошибку, потеряв, допустим, fi). Однако в целом он достаточно понятен, запутаться, как правило, довольно сложно (во многом как раз благодаря нагруженному синтаксису – всё очень наглядно, не запутаешься в фигурных скобках, как это возможно в Си).

4 Выводы

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