Отчёта по лабораторной работе №13

Программирование в командномпроцессоре ОС UNIX. Расширенное программирование

Джумаев Бегенч

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 кон-струкций и циклов

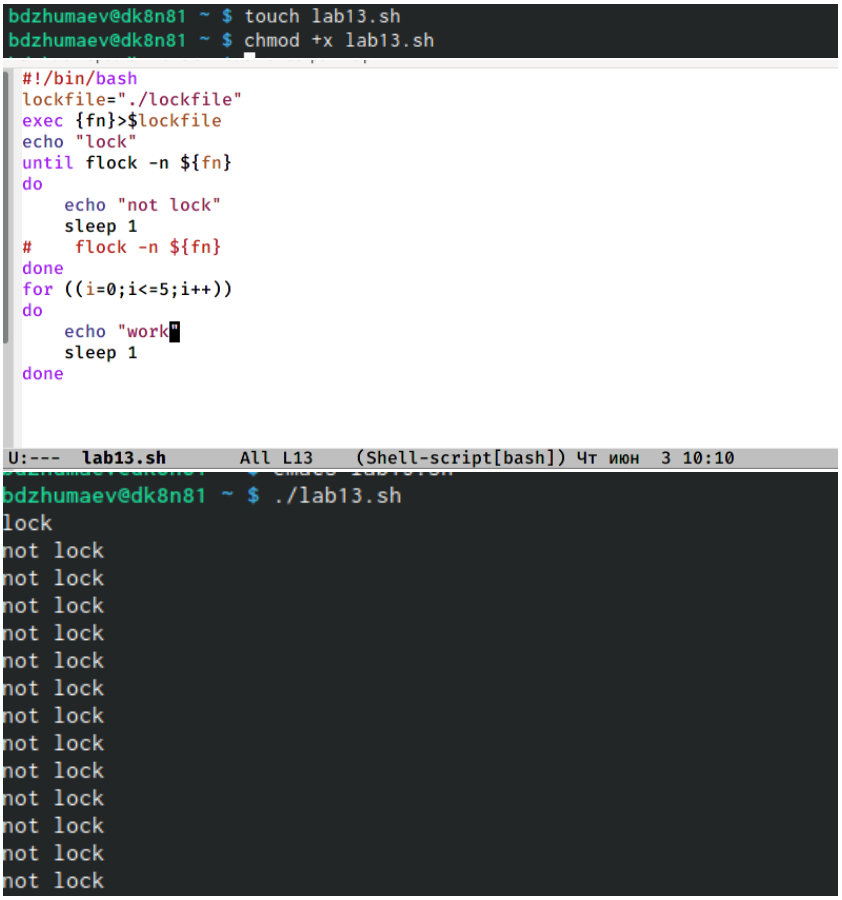
2 Задание

1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени t1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени t2<>t1, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустить командный файл в одном виртуальном терминале в фоновомрежиме, перенаправив его вывод в другой (> /dev/tty#, где#— номер терминала куда перенаправляется вывод), в котором также запущен этот файл, ноне фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов.
2. Реализовать командуmanc помощью командного файла. Изучите содержимоекаталога/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 должен получать в виде аргумента командойстроки название команды и в виде результата выдавать справку об этой коман-де или сообщение об отсутствии справки, если соответствующего файла нет в каталоге man1.
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

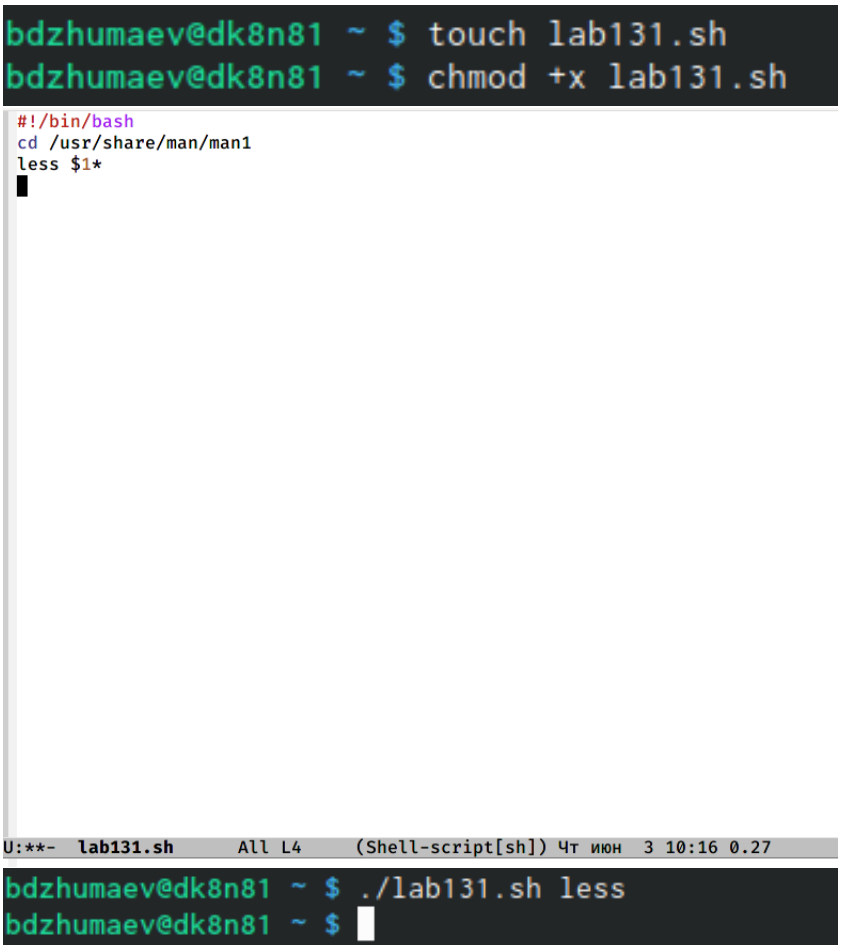
32767.

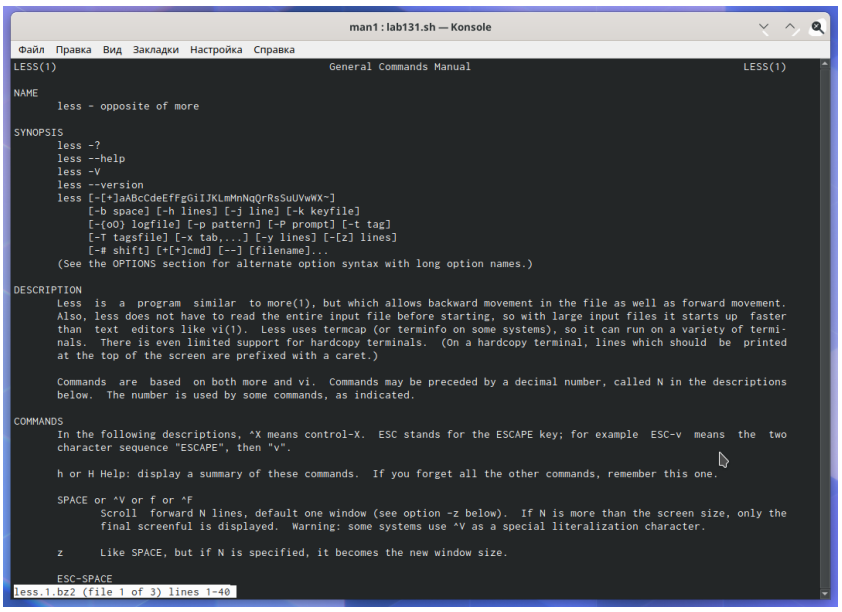
3 Выполнение лабораторной работы

1. Написал командный файл, реализующий упрощённый механизм семафоров. Я запустил командный файл в одном вертуальном 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. Я доработал программу так, чтобы имелась возможность взаимодействия трёх и более процессов.

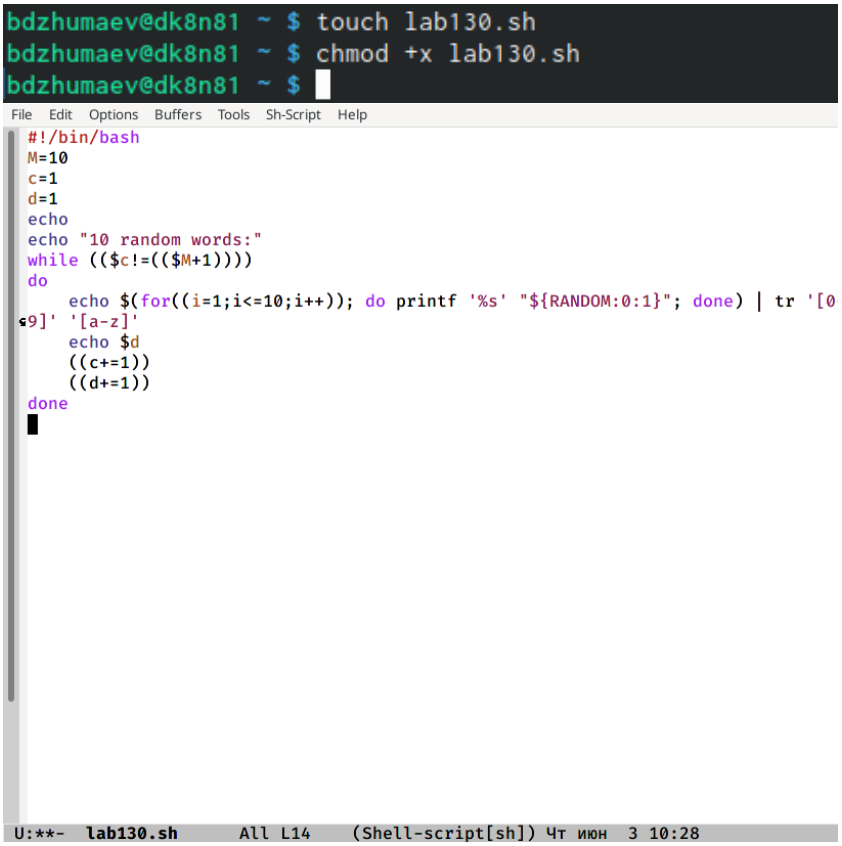


2. Я реализовал команду manc помощью командного файла.Изучал содержимо екаталога/usr/share/man/man1. Командный файл должен получать в виде аргумента команднойстроки название команды и в виде результата выдавать справку об этой коман-де или сообщение об отсутствии справки, если соответствующего файла нет в каталоге man1.





3. Я используя встроенную переменную $RANDOM, написал командный файл, генерирующий случайную последовательность букв латинского алфавита.




bdzhumaev@dk8n81 ~ $ ./lab130.sh
10 random words:
bbbdcchcbb
1
ccccedcccc
2
ececcbcbfb
3
bbddjcdccc
4
dcebbbbccd
5
dibbbjdbcb
6
cggdccbbcc
7
dbdbcjccdj
8
hccjebbecb
9
fbcbbcfidh
10
bdzhumaev@dk8n81 ~ $

4 Выводы

Я изучал основы программирования в оболочке ОС UNIX и научил писать более сложнее командные файлы с использованием логических управляющих и циклов.

5 Контрольные вопросы

1. В строке while [$1 != “exit”] квадратные скобки надо заменить на круглые.

2. Есть несколько видов конкатенации строк. Например,

VAR1=“Hello,”
VAR2=” World”
VAR3=“$VAR1$VAR2”
echo “$VAR3”

3. Команда seq выводит последовательность целых или действительных чисел, подходящую для передачи в другие программы. В bash можно использовать seq с циклом for, используя подстановку команд. Например,

$ for i in $(seq 1 0.5 4)
do
echo “The number is $i”
done

4. Результатом вычисления выражения $((10/3)) будет число 3.

5. Список того, что можно получить, используя Z Shell вместо Bash:

Встроенная команда zmv поможет массово переименовать файлы/директории, например, чтобы добавить ‘.txt’ к имени каждого файла, запустите zmv –C ’(*)(#q.)’ ‘$1.txt’.

Утилита zcalc — это замечательный калькулятор командной строки, удобный способ считать быстро, не покидая терминал.

Команда zparseopts — это однострочник, который поможет разобрать сложные варианты, которые предоставляются скрипту.

Команда autopushd позволяет делать popd после того, как с помощью cd, чтобы вернуться в предыдущую директорию.

Поддержка чисел с плавающей точкой (коей Bash не содержит).

Поддержка для структур данных «хэш».