

Лабораторная работа №12

Операционные системы

Кавказова Диана Алексеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Контрольные вопросы	10
4	Выводы	11
	Список литературы	12

Список иллюстраций

2.1	рисунок	7
2.2	рисунок	8
2.3	рисунок	9

Список таблиц

1 Цель работы

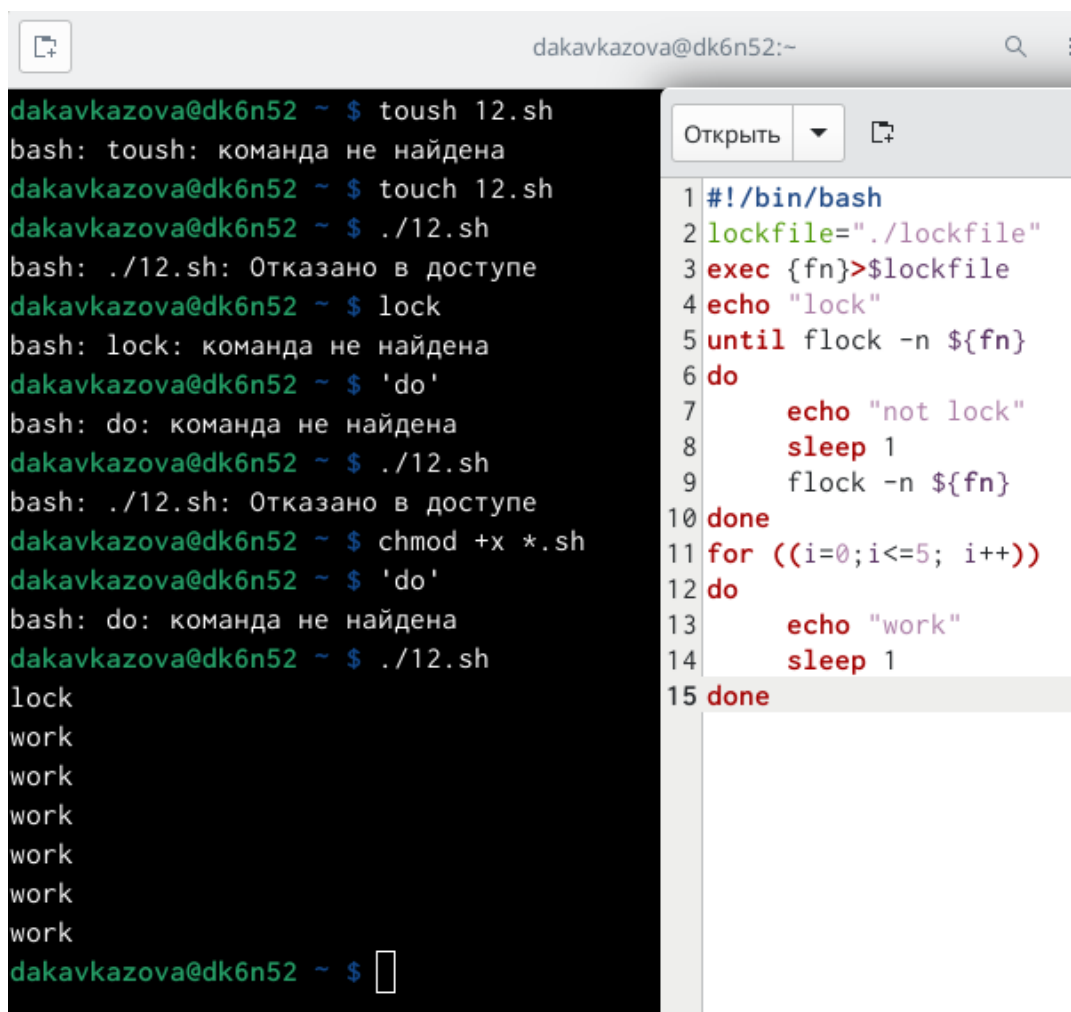
Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

2 Задание

1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный 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 (процессом). 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 фоновом, а в привилегированном режиме. Доработать программу так, чтобы имела возможность взаимодействия трёх и более процессов.
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 команд. 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`, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита.

Учтите, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767. # Выполнение лабораторной работы

4. Написали 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). Запустили командный файл.



```
dakavkazova@dk6n52 ~ $ touch 12.sh
bash: touch: команда не найдена
dakavkazova@dk6n52 ~ $ touch 12.sh
dakavkazova@dk6n52 ~ $ ./12.sh
bash: ./12.sh: Отказано в доступе
dakavkazova@dk6n52 ~ $ lock
bash: lock: команда не найдена
dakavkazova@dk6n52 ~ $ 'do'
bash: do: команда не найдена
dakavkazova@dk6n52 ~ $ ./12.sh
bash: ./12.sh: Отказано в доступе
dakavkazova@dk6n52 ~ $ chmod +x *.sh
dakavkazova@dk6n52 ~ $ 'do'
bash: do: команда не найдена
dakavkazova@dk6n52 ~ $ ./12.sh
lock
work
work
work
work
work
work
dakavkazova@dk6n52 ~ $
```

```
1 #!/bin/bash
2 lockfile="./lockfile"
3 exec {fn}>$lockfile
4 echo "lock"
5 until flock -n ${fn}
6 do
7     echo "not lock"
8     sleep 1
9     flock -n ${fn}
10 done
11 for ((i=0;i<=5; i++))
12 do
13     echo "work"
14     sleep 1
15 done
```

Рис. 2.1: рисунок

2. Реализовали 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 команд. 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`.

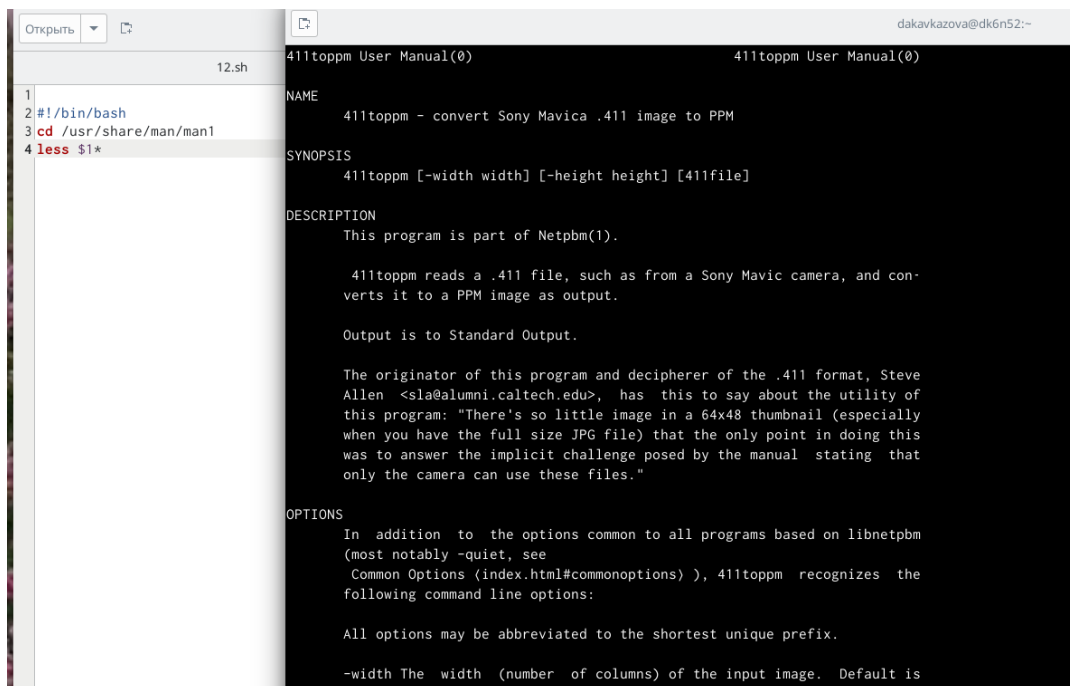


Рис. 2.2: рисунок

3. Используя встроенную переменную `$RANDOM`, написали командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что `$RANDOM` выдаёт псевдослучайные числа в диапазоне от 0 до 32767.


```
12-2.sh
dakavkazova@dk6n52:~
12.sh
12-1.sh
1 #!/bin/bash
2 M=10
3 c=1
4 d=1
5 echo
6 echo "10 random words:"
7 while (($c!=($M+1)))
8 do
9     echo $(for((i=1;i<=10;i++)); do printf '%s'
10     echo $d
11     ((c+=1))
12     ((d+=1))
13 done
work
work
work
work
work
work
dakavkazova@dk6n52 ~ $ touch 12-1.sh
dakavkazova@dk6n52 ~ $ chmod +x *.sh
dakavkazova@dk6n52 ~ $ ./12-1.sh
dakavkazova@dk6n52 ~ $ ./12-1.sh
dakavkazova@dk6n52 ~ $ touch 12-2.sh
dakavkazova@dk6n52 ~ $ chmod +x *.sh
dakavkazova@dk6n52 ~ $ ./12-2.sh
10 random words:
bggcdfjdc
1
bbcbcbdbb
2
cbdcdbccbd
3
bbcibdbjcj
4
ccccbcdicb
5
gdbjiccbec
6
cbbdbcbde
7
cccefbccbb
8
cccbcbbec
9
dhgbfcdccb
10
```

Рис. 2.3: рисунок

3 Контрольные вопросы

1. Объясните понятие командной оболочки. Приведите примеры командных оболочек. Чем они отличаются?
2. Что такое POSIX?
3. Как определяются переменные и массивы в языке программирования bash?
4. Каково назначение операторов let и read?
5. Какие арифметические операции можно применять в языке программирования bash?
6. Что означает операция (())?
7. Какие стандартные имена переменных Вам известны?
8. Что такое метасимволы?
9. Как экранировать метасимволы?
10. Как создавать и запускать командные файлы?
11. Как определяются функции в языке программирования bash?
12. Каким образом можно выяснить, является файл каталогом или обычным файлом?
13. Каково назначение команд set, typeset и unset?
14. Как передаются параметры в командные файлы?
15. Назовите специальные переменные языка bash и их назначение.

4 Выводы

Изучили основы программирования в оболочке ОС UNIX/Linux. Научились писать небольшие командные файлы.

Список литературы