

Лабораторная работа 12

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

Коршунова Полина

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 3 |
| 2 | Задание | 4 |
| 3 | Теоретическое введение | 5 |
| 4 | Выполнение лабораторной работы | 6 |
| 5 | Вывод | 10 |
| 6 | Контрольные вопросы | 11 |

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

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, в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась 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.

3 Теоретическое введение

Преимущества и недостатки Bash:

Многие языки программирования намного удобнее и понятнее для пользователя. Например, Python более быстр, так как компилируется байтами. Однако главное преимущество Bash – его повсеместное распространение. Более того, Bash позволяет очень легко работать с файловой системой без лишних конструкций (в отличие от других языков программирования). Но относительно таких bash очень сжат. То есть, например, C имеет гораздо более широкие возможности для разработчика.

4 Выполнение лабораторной работы

1. Я написала командный файл, реализующий упрощенный механизм семафоров.

```
lockfile="./locking.file"

exec {fn}>"$lockfile"
if test -f "$lockfile"
then
    while [ 1 != 0 ]
    do
        if flock -n ${fh}
        then
            echo "file was locked"
            sleep 4
            echo "unlocking"
            flock -u ${fn}

        else
            echo "file was unlocked"
            sleep 3
        fi
    done
fi
```

Рис. 4.1: Скрипт к заданию 1.

Затем я добавила право на исполнение файла и выполнила его.

1

```

pykorshunova@dk8n52:~
command=""

while getopts :c: opt
do
case $opt in
    c)command="$OPTARG";;
esac
done

if test -f "/usr/share/man/man1/$command.1.gz"
then less /usr/share/man/man1/$command.1.gz
else
echo "no such a command!"
fi
~

```

Рис. 4.4: Скрипт к заданию 2.

Затем я добавила право на исполнение файла и выполнила его.

```

pykorshunova@dk8n52:~
pykorshunova@dk8n52 ~ $ vi lab12_2
pykorshunova@dk8n52 ~ $ chmod +x lab12_2
pykorshunova@dk8n52 ~ $ ./lab12_2
no such a command!
pykorshunova@dk8n52 ~ $ 

```

Рис. 4.5: Результат выполнения скрипта 2.

3. Я написала командный файл, который генерировал случайную последовательность букв латинского алфавита, для этого я использовала встроенную переменную \$RANDOM.


```
pykorshunova
echo $RANDOM | tr '0-9' 'a-zA-Z'
```

Рис. 4.6: Скрипт к заданию 3.

Затем я добавила право на исполнение файла и выполнила его.

```
pykorshunova@dk8n52 ~ $ vi lab12_3
pykorshunova@dk8n52 ~ $ chmod +x lab12_3
pykorshunova@dk8n52 ~ $ ./lab12_3
dbaec
pykorshunova@dk8n52 ~ $
```

Рис. 4.7: Результат выполнения скрипта 3.

5 Вывод

В ходе выполнения лабораторной работы я изучила основы программирования в оболочке ОС UNIX, а также научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

6 Контрольные вопросы

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

Между скобками должны быть пробелы, иначе символы в скобках и сами скобки будут восприняты как один элемент.

2. Как объединить (конкатенация) несколько строк в одну?

```
cat file.txt | xargs
```

3. Найдите информацию об утилите `seq`. Какими иными способами можно реализовать её функционал при программировании на `bash`?

Команда `seq` выводит последовательность целых или действительных чисел, подходящую для передачи в другие программы. Реализовать ее функционал можно командой `for n in {1..5} do done`

4. Какой результат даст вычисление выражения `$((10/3))`? Вычисление этого выражения даст результат 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. Если вы собираетесь писать скрипт, который будет запускать множество разработчиков, то рекомендуется `Bash`. Если скрипты вам не нужны - `Zsh`.

6. Проверьте, верен ли синтаксис данной конструкции `1 for ((a=1; a <= LIMIT; a++))`

Да, этот синтаксис верен.

7. Сравните язык `bash` с какими-либо языками программирования. Какие преимущества у `bash` по сравнению с ними? Какие недостатки?

Многие языки программирования намного удобнее и понятнее для пользователя. Например, Python более быстр, так как компилируется байтами. Однако главное преимущество Bash – его повсеместное распространение. Более того, Bash позволяет очень легко работать с файловой системой без лишних конструкций (в отличие от других языков программирования). Но относительно таких `bash` очень сжат. То есть, например, C имеет гораздо более широкие возможности для разработчика.