Операционные системы

Лабораторная работа №12

Матюшкин Денис Владимирович (НПИбд-02-21)

Содержание

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. Напишем командный файл, реализующий упрощённый механизм семафоров. Командный файл в течение некоторого времени t1 дожидается освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использует его в течение некоторого времени t2<>t1, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустим 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. Доработаем программу так, чтобы имелась возможность взаимодействия трёх и более процессов (рис. 2.1). Проверим написанный командный файл (рис. 2.2).

```
#!/bin/bash

lockfile="./lock.file"
exec {fn}>$lockfile

if test -f "$lockfile"
then
    while [ 1 = 1 ]
    do
        if flock -n ${fn}
        then
            echo "Файл заблокирован"
            sleep 5
            echo "Файл разблокирован"
            flock -u ${fn}
        else
            echo "Файл уже заблокирован"
            sleep 5
        fi
    done
else
    echo "Файл не найден"
fi
```

Рис. 2.1: Командный файл

```
[dvmatyushkin@dvmatyushkin ~]$ chmod +x lab12_1.sh
[dvmatyushkin@dvmatyushkin ~]$ ./lab12_1.sh
Файл заблокирован
Файл разблокирован
Файл заблокирован
Файл разблокирован
Файл заблокирован
Файл разблокирован
Файл заблокирован
Файл разблокирован
Файл заблокирован
Файл разблокирован
```

Рис. 2.2: Проверка файла

2. Реализуем команду man с помощью командного файла. Изучим 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 выдает справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге man1 (рис. 2.3). Проверим написанный командный

файл (рис. 2.4 и рис. 2.5).

```
#!/bin/bash

name=""

while getopts "n:" opt
do
    case $opt in
        n)name=$OPTARG;;
    esac
done
if test -f "/usr/share/man/man1/$name.1.gz"
then
    less /usr/share/man/man1/"$name".1.gz
else
    echo "Такая команда не найдена!"
fi
```

Рис. 2.3: Командный файл

```
[dvmatyushkin@dvmatyushkin ~]$ chmod +x lab12_2.sh
[dvmatyushkin@dvmatyushkin ~]$ ./lab12_2.sh -n mkdir
[dvmatyushkin@dvmatyushkin ~]$
```

Рис. 2.4: Проверка файла

```
MKDIR(1)                                          User Commands

ESC[1mNAMEESC[0m
       mkdir - make directories

ESC[1mSYNOPSISESC[0m
       ESC[1mmkdir ESC[22m[ESC[4mOPTIONESC[24m]... ESC[4mDIRECTORYESC[24m...

ESC[1mDESCRIPTIONESC[0m
       Create the DIRECTORY(ies), if they do not already exist.
```

Рис. 2.5: Проверка файла

3. Используя встроенную переменную $RANDOM, напишем командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтем, что $RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767 (рис. 2.6). Проверим написанный командный файл (рис. 2.7).

```
#!/bin/bash

echo $RANDOM | tr '[0-9]' '[a-zA-z]'
```

Рис. 2.6: Командный файл

```
[dvmatyushkin@dvmatyushkin ~]$ emacs lab12_3.sh
[dvmatyushkin@dvmatyushkin ~]$ chmod +x lab12_3.sh
[dvmatyushkin@dvmatyushkin ~]$ ./lab12_3.sh
gcbb
[dvmatyushkin@dvmatyushkin ~]$ ./lab12_3.sh
bhjci
[dvmatyushkin@dvmatyushkin ~]$ ./lab12_3.sh
cfafb
[dvmatyushkin@dvmatyushkin ~]$ ./lab12_3.sh
daejh
[dvmatyushkin@dvmatyushkin ~]$
```

Рис. 2.7: Проверка файла

3 Контрольные вопросы

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

Ответ: $1. Так же 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 …

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 <КОМАНДА>
done
```

4. Какой результат даст вычисление выражения $((10/3))?

Ответ: 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 (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++))

Ответ: 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 отличи от обычного языка программирования). Но относительно обычных языков программирования bash очень сжат. Тот же Си имеет гораздо более широкие возможности для разработчика.

4 Вывод

- В ходе этой лабораторной работы мы изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.