

Лабораторная работа No 12

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

Поляков Глеб Сергеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Выводы	10
	Список литературы	11

Список иллюстраций

Список таблиц

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 (`>/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 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 Выполнение лабораторной работы

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 имелась возможность взаимодействия трёх и более процессов.

```
#!/bin/bash
semaphore=/tmp/semaphore.lock
t1=10
t2=20

echo "Process $$: waiting for semaphore..."

while true; do
    if ( set -o noclobber; echo "$$" > "$semaphore" ) 2> /dev/null; then
        echo "Process $$: semaphore acquired, using resource for $t2 seconds..."
        sleep $t2
        rm "$semaphore"
        echo "Process $$: semaphore released."
        break
    else
        echo "Process $$: semaphore busy, waiting $t1 seconds..."
        sleep $t1
    fi
done
```

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.

```
1 #!/bin/bash
2
3 if [ -z "$1" ]; then
4     echo "Usage: man command"
5     exit 1
6 fi
7
8 manpage=/usr/share/man/man1/$1.1.gz
9
10 if [ -f "$manpage" ]; then
11     zcat "$manpage" | less
12 else
13     echo "No manual entry for $1"
14     exit 1
15 fi
```

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

```
1 #!/bin/bash
2
3 semaphore=/tmp/semaphore.lock
4 t1=10
5 t2=20
6
7 echo "Process $$: waiting for semaphore..."
8
9 while true; do
10     if ( set -o noclobber; echo "$$" > "$semaphore" ) 2> /dev/null; then
11         echo "Process $$: semaphore acquired, using resource for $t2 seconds..."
12         sleep $t2
13         rm "$semaphore"
14         echo "Process $$: semaphore released."
15         break
16     else
17         echo "Process $$: semaphore busy, waiting $t1 seconds..."
18         sleep $t1
19     fi
20 done
```

32767.

4 Выводы

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

Список литературы