

Лабораторная работа №14

операционные системы

Лисенков Е.Р.

Российский университет дружбы народов, Москва, Россия

Информация

- Лисенков Егор Романович
- студент
- Российский университет дружбы народов
- 1132232881@rudn.ru
- <https://github.com/erlisenkov>



Вводная часть

Цель данной лабораторной работы - изучить основы программирования в оболочке ОС UNIX, научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Выполнение лабораторной работы

Выполнение лабораторной работы

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



```
121.sh
~/work/os/lab14

Открыть ▼ + Сохранить

1 #!/bin/bash
2
3 lockfile="./lock.file"
4 exec {fn}>$lockfile
5
6 while test -f "$lockfile"
7 do
8   if flock -n ${fn}
9   then
10     echo "File is blocked"
11     sleep 5
12     echo "File is unlocked"
13     flock -u ${fn}
14   else
15     echo "File is blocked"
16     sleep 5
17   fi
18 done
```

Рис. 1: Код программы

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

```
#!/bin/bash
```

```
a=$1  
if test -f "/usr/share/man/man1/$a.1.gz"  
then less /usr/share/man/man1/$a.1.gz  
else  
echo "There is no such command"  
fi
```



The image shows a terminal window with a title bar. The title bar contains the text '122.sh' and the path '~/.work/os/lab14'. On the left side of the title bar are buttons for 'Открыть' (Open) and a dropdown arrow. On the right side are buttons for 'Сохранить' (Save), a menu icon (three horizontal lines), and a close icon (an 'x'). The terminal content shows a shell script with line numbers 1 through 8. The script starts with a shebang '1 #! /bin/bash', followed by a blank line, then '3 a=\$1'. Line 4 is '4 if test -f "/usr/share/man/man1/\$a.1.gz"', line 5 is '5 then less /usr/share/man/man1/\$a.1.gz', line 6 is '6 else', line 7 is '7 echo "There is no such command"', and line 8 is '8 fi'.

```
1 #! /bin/bash
2
3 a=$1
4 if test -f "/usr/share/man/man1/$a.1.gz"
5 then less /usr/share/man/man1/$a.1.gz
6 else
7 echo "There is no such command"
8 fi
```

Последняя программа выглядит иначе.

Открыть 123.sh Сохранить x

~/work/os/lab14

```
1 #!/bin/bash
2
3 a=$1
4
5 for ((i=0; i<$a; i++))
6 do
7     ((char=$RANDOM%26+1))
8     case $char in
9         1) echo -n a;; 2) echo -n b;; 3) echo -n c;; 4) echo -n d;; 5) echo -n e;; 6) echo -n f;;
10        7) echo -n g;; 8) echo -n h;; 9) echo -n i;; 10) echo -n j;; 11) echo -n k;; 12) echo -n l;;
11        13) echo -n m;; 14) echo -n n;; 15) echo -n o;; 16) echo -n p;; 17) echo -n r;; 18) echo -n s;;
12        19) echo -n t;; 20) echo -n q;; 21) echo -n u;; 22) echo -n v;;
13        23) echo -n w;; 24) echo -n x;; 25) echo -n y;; 26) echo -n z;;
14    esac
15 done
16 echo
```

Выводы

Я усвоил материал и готов к дальнейшему изучению линукс!

Ответы на контрольные вопросы

Каково предназначение команды `getopts`?

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

В данной строчке допущены следующие ошибки: не хватает пробелов после первой скобки `[` и перед второй скобкой `]` выражение `$1` необходимо взять в `"`, потому что эта переменная может содержать пробелы Таким образом, правильный вариант должен выглядеть так: `while ["$1" != "exit"]`

Как объединить (конкатенация) несколько строк в одну?

Чтобы объединить несколько строк в одну, можно воспользоваться несколькими способами:

Первый: VAR1="Hello," VAR2=" World" VAR3="\$VAR1\$VAR2" echo "\$VAR3" Результат: Hello, World

Второй: VAR1="Hello," VAR1+=" World" echo "\$VAR1" Результат: Hello, World

Какие операторы управления действиями вы знаете?

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

Команда `seq` в Linux используется для генерации чисел от ПЕРВОГО до ПОСЛЕДНЕГО шага INCREMENT. Параметры: `seq LAST`: если задан только один аргумент, он создает числа от 1 до LAST с шагом шага, равным 1. Если LAST меньше 1, значение `is` не выдает. `seq FIRST LAST`: когда заданы два аргумента, он генерирует числа от FIRST до LAST с шагом 1, равным 1. Если LAST меньше FIRST, он не выдает никаких выходных данных. `seq FIRST INCREMENT LAST`: когда заданы три аргумента, он генерирует числа от FIRST до LAST на шаге INCREMENT. Если LAST меньше, чем FIRST, он не производит вывод. `seq -f «FORMAT» FIRST INCREMENT LAST`: эта команда используется для генерации последовательности в форматированном виде. FIRST и INCREMENT являются необязательными. `seq -s «STRING» ПЕРВЫЙ ВКЛЮЧЕНО`: Эта команда используется для STRING для разделения чисел. По умолчанию это значение равно `/n`. FIRST

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

Результатом данного выражения $\$(10/3)$ будет 3, потому что это целочисленное деление без остатка.

Укажите кратко основные отличия командной оболочки zsh от bash.

Отличия командной оболочки zsh от bash: В zsh более быстрое автодополнение для cd с помощью Tab В zsh существует калькулятор zcalc, способный выполнять вычисления внутри терминала В zsh поддерживаются числа с плавающей запятой В zsh поддерживаются структуры данных «хэш» В zsh поддерживается раскрытие полного пути на основе неполных данных В zsh поддерживается замена части пути В zsh есть возможность отображать разделенный экран, такой же как разделенный экран vim

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

`for ((a=1; a <= LIMIT; a++))` синтаксис данной конструкции верен, потому что, используя двойные круглые скобки, можно не писать `$` перед переменными `()`.

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

Преимущества и недостатки скриптового языка `bash`:

- Один из самых распространенных и ставится по умолчанию в большинстве дистрибутивах Linux, MacOS
 - Удобное перенаправление ввода/вывода
 - Большое количество команд для работы с файловыми системами Linux
 - Можно писать собственные скрипты, упрощающие работу в Linux
- Недостатки скриптового языка `bash`:
- Дополнительные библиотеки других языков позволяют выполнить больше действий
 - Bash не является языком общего назначения
 - Утилиты, при выполнении скрипта, запускают свои процессы, которые, в свою очередь, отражаются на скорости выполнения этого скрипта