

Лабораторная работа №14

Операционные системы

Чистов Д. М.

11 Мая 2024

Российский университет дружбы народов, Москва, Россия

Вступительная информация

Цель работы

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

Задания

1. Программа 1
2. Программа 2
3. Программа 3

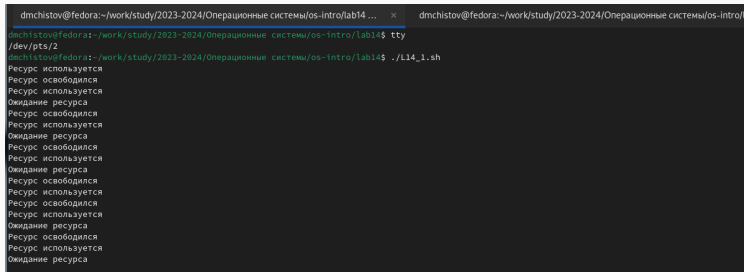
Выполнение лабораторной работы

Программа 1

Задание: “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.

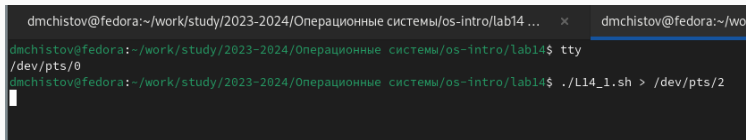
Программа 1

Приступаю к выполнению работы. Создаю файл с кодом и пишу код (сама программа будет показана позже), выполняю - готово.



```
dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14 ... x dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/l
dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14$ tty
/dev/pts/2
dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14$ ./L14_1.sh
Ресурс используется
Ресурс освобожден
Ресурс используется
Ожидание ресурса
Ресурс освобожден
Ресурс используется
Ожидание ресурса
Ресурс освобожден
Ресурс используется
Ожидание ресурса
Ресурс освобожден
Ресурс используется
Ресурс освобожден
Ресурс используется
Ожидание ресурса
Ресурс освобожден
Ресурс используется
Ожидание ресурса
```

Рис. 1: Задание 1 - выполнено вид из терминала 1

A terminal window with a dark background and light green text. The window title bar shows two tabs: 'dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14 ...' and 'dmchistov@fedora:~/wo'. The terminal content shows the user 'dmchistov@fedora' at the directory '~/work/study/2023-2024/Операционные системы/os-intro/lab14' running the command 'tty', which outputs '/dev/pts/0'. Then, the user runs './L14_1.sh > /dev/pts/2', and a white cursor is visible on the next line.

```
dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14$ tty
/dev/pts/0
dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14$ ./L14_1.sh > /dev/pts/2
█
```

Рис. 2: Задание 1 - выполнено вид из терминала 2

Программа 1

Код программы 1: создаём файл, и запускаем его. Бесконечно идём циклом `while` и проверяем утилитой `flock`, если программа используется, говорим об этом, ждём 5 секунд и освобождаем ресурс. Если не используется, то пишем, что ожидаем, и ждём 8 секунд.

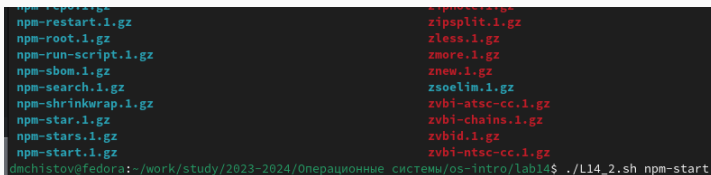
```
1  #!/bin/bash
2
3  file="./.lock.file"
4  exec {fn}<>$file
5
6  while true
7  do
8      if flock -n ${fn}; then
9          echo "Ресурс используется"
10         sleep 5
11         flock -u ${fn}
12         echo "Ресурс освобожден"
13         sleep 8
```

Программа 2

Задание: “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`.”

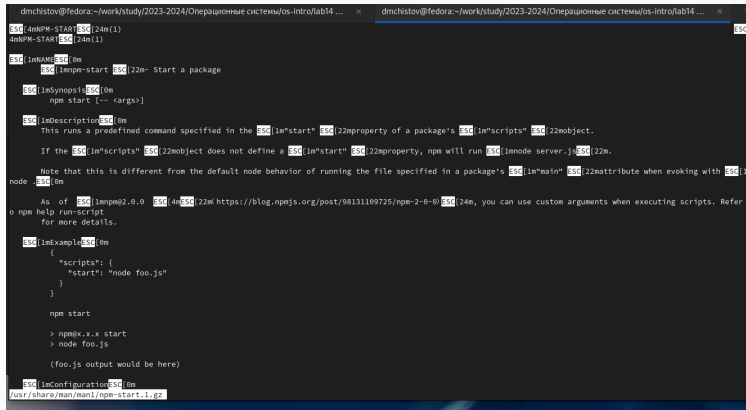
Программа 2

Приступаю к выполнению - создаю, даю права, пишу код, исправляю ошибки, запускаю - работает



```
npm-restart.1.gz  
npm-root.1.gz  
npm-run-script.1.gz  
npm-sbom.1.gz  
npm-search.1.gz  
npm-shrinkwrap.1.gz  
npm-star.1.gz  
npm-stars.1.gz  
npm-start.1.gz  
zipncat.1.gz  
zipsplit.1.gz  
zless.1.gz  
zmore.1.gz  
znew.1.gz  
zsoelim.1.gz  
zvbi-atsc-cc.1.gz  
zvbi-chains.1.gz  
zvbid.1.gz  
zvbi-ntsc-cc.1.gz  
dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14$ ./L14_2.sh npm-start
```

Рис. 4: Задание 2 - выполнено



```
dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14 ... x  dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14 ... x
ESC[4mNPM-STARTESC[24m(1)
4mNPM-STARTESC[24m(1)

ESC[1mNAMEESC[0m
ESC[1mnpn-startESC[22m- Start a package

ESC[1mSynopsisESC[0m
  npn start [-- <args>]

ESC[1mDescriptionESC[0m
  This runs a predefined command specified in the ESC[1m"start"ESC[22mproperty of a package's ESC[1m"scripts"ESC[22mobject.

  If the ESC[1m"scripts"ESC[22mobject does not define a ESC[1m"start"ESC[22mproperty, npn will run ESC[1mnode server.jsESC[22m.

  Note that this is different from the default node behavior of running the file specified in a package's ESC[1m"main"ESC[22mattribute when evoking with ESC[1mnodeESC[0m.

  As of ESC[1mnpn@2.0.0ESC[0m ESC[4mESC[22mhttps://blog.npmjs.org/post/98131109725/npm-2-0-0ESC[24m, you can use custom arguments when executing scripts. Refer
o npn help run-script
  for more details.

ESC[1mExampleESC[0m
{
  "scripts": {
    "start": "node foo.js"
  }
}

npn start

> npn@x.x.x start
> node foo.js

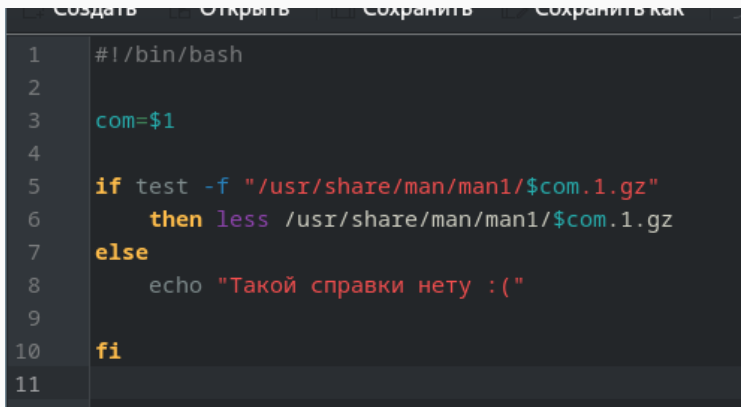
(foo.js output would be here)

ESC[1mConfigurationESC[0m
/usr/share/man/man1/npn-start.1.gz
```

Рис. 5: Задание 2 - выполнено

Программа 2

Код программы 2: Создаю переменную (название команды, которую мы хотим изучить), утилитой test проверяем, есть ли документация по этой команде в заданной директории. Если есть, то читаем её утилитой less, если нету, пишем что такой документации нет.

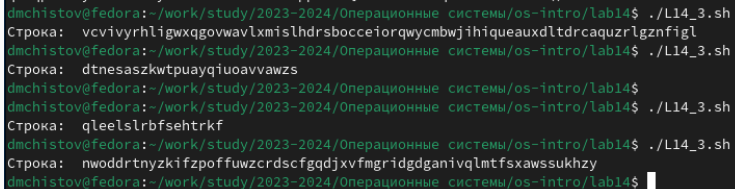


```
1  #!/bin/bash
2
3  com=$1
4
5  if test -f "/usr/share/man/man1/$com.1.gz"
6      then less /usr/share/man/man1/$com.1.gz
7  else
8      echo "Такой справки нету :("
9
10 fi
11
```

Программа 3

Задание: “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.”

Создаю файл, даю права, начинаю писать код, всё работает



```
dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14$ ./L14_3.sh
Строка: vcvivyrhligwxqgovwavlxmislhdrsboceiorqwyсmbwjihiqueauxdltdrcaquzrlgznfigl
dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14$ ./L14_3.sh
Строка: dtnesaszkwtpuayqiuoavvawzs
dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14$
dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14$ ./L14_3.sh
Строка: qleelslrbfsehtrkf
dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14$ ./L14_3.sh
Строка: nwoddrtnyzkifzpoffuwzcrdscfgqdxvfmggridgdganivqlmtfsxawssukhzy
dmchistov@fedora:~/work/study/2023-2024/Операционные системы/os-intro/lab14$
```

Рис. 7: Задание 3 - выполнено

Код программы 3: Создаю переменную - строку со всеми буквами латинского алфавита. Затем задаю ещё одну переменную “Длина строки” - утилитой \$RANDOM выбираю случайное число. Это и будет длина. Задаю переменную результата - пустую строку. Циклом for иду столько число раз, сколько в перменной “длина” В каждой итерации - беру индекс - утилитой \$RANDOM это будет случайное число до 26. Затем временной переменной letter достаю из строки из алфавита букву, которая находится по ранее найденному индексу. Приписываю эту букву к перменной результат. Вывожу.

Программа 3

```
1  #!/bin/bash
2
3  alphabet="abcdefghijklmnopqrstuvwxyz"
4
5  length=$(( $RANDOM % 100 + 1 ))
6
7  result=""
8
9  for ((i=0; i<length; i++))
10 do
11     index=$(( $RANDOM % 26 ))
12     letter="${alphabet:$index:1}"
13     result+="$letter"
14 done
15
16 echo "Строка: " $result
17
18
```

Выводы

В результате выполнения данной работы я изучил основы программирования в оболочке ОС UNIX/Linux. Научился писать небольшие командные файлы.

Список литературы

Лабораторная работы №14

Работа с строками и shell scripting

Команда flock