
Front matter

lang: ru-RU title: "Программирование в командном процессоре ОС UNIX. Расширенное программирование" author: | Petrov Artyem E.\inst{1} \and institute: | \inst{1}RUDN University, Moscow, Russian Federation \and date: NEC--2022, 25 May, 2022 Moscow, the Russian Federation

Formatting

toc: false slide_level: 2 theme: metropolis header-includes:

- \metroset{progressbar=frametitle,sectionpage=progressbar,numbering=fraction}
- 'makeatletter'
- 'beamer@ignorenonframefalse'
- 'makeatother' aspectratio: 43 section-titles: true

Цель работы

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

Задание

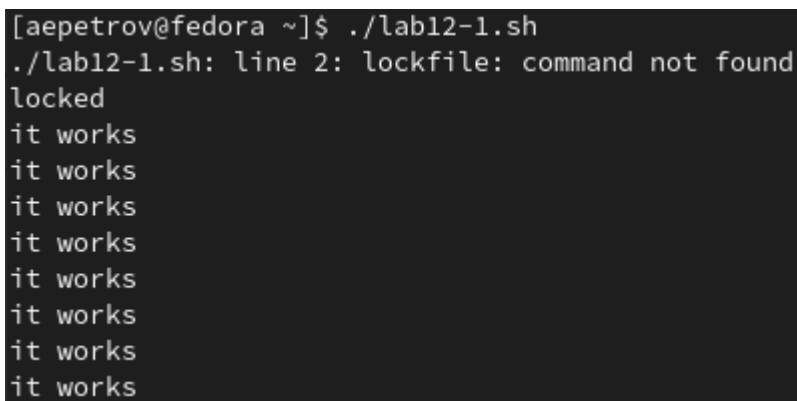
1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени t_1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени $t_2 < t_1$, также выдавая информацию 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.

Выполнение лабораторной работы

Задание 1. Рис. [-@fig:001].

```
#!/bin/bash
lockfile = "./lock.file"
exec {fn}>lockfile

echo 'locked'
until flock -n ${fn}
do
    echo "idk not wrkin"
    sleep 1
    flick -n ${fn}
done
for ((i=0; i<=7; i++))
do
    echo "it works"
    sleep 1
done
```

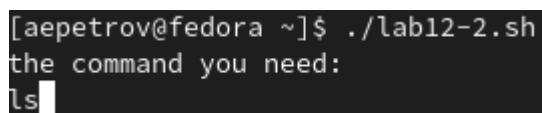


```
[aepetrov@fedora ~]$ ./lab12-1.sh
./lab12-1.sh: line 2: lockfile: command not found
locked
it works
it works
it works
it works
it works
it works
it works
it works
it works
```

{ #fig:001 width=70% }

Задание 2. Рис. [-@fig:002]-[-@fig:003].

```
#!/bin/bash
cd /usr/share/man/man1
echo the command you need:
read command
less $command*
```



```
[aepetrov@fedora ~]$ ./lab12-2.sh
the command you need:
ls
```

{ #fig:002 width=70% }

```

LS(1)                                     User Commands                                     LS(1)

ESC[1mNAMEESC[0m
    ls - list directory contents

ESC[1mSYNOPSISESC[0m
    ESC[1mls ESC[22m[ESC[4mOPTIONESC[24m]... [ESC[4mFILEESC[24m]...

ESC[1mDESCRIPTIONESC[0m
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of ESC[1m-cftuvSUX ESC[22mnor ESC[1m-
-sort ESC[22mis speci-
    fied.

    Mandatory arguments to long options are mandatory for short options
    too.

    ESC[1m-aESC[22m, ESC[1m--allESC[0m
        do not ignore entries starting with .

    ESC[1m-AESC[22m, ESC[1m--almost-allESC[0m
        do not list implied . and ..

ls 1.gz (file 1 of 10)

```

#fig:003 width=70% }

Задание 3. Рис. [-@fig:004].

```

#!/bin/bash
num
echo enter num of random stuff:
read word
for((i=0;i<$word;i++))
do
    echo $RANDOM | tr '[0-32670]' '[a-z]'
done

```

```

[aepetrov@fedora ~]$ ./lab12-3.sh
./lab12-3.sh: line 2: num: command not found
enter num of random stuff:
10
bff
e44g9
bg9be
e8e9
bfd98
948e
b5he4
gf54
e55e4
be48g

```

{ #fig:004 width=70% }

Выводы

Благодаря данной лабораторной работе я научился создавать генератор случайных последовательностей символов, написал упрощенный механизм семафоров и узнал, как реализовать замену для базовых команды Unix