

Лабораторная работа н.12

Программирование в командном процессоре ОС UNIX. Расширенное
программирование

Петров Артем Евгеньевич

Содержание

Цель работы	4
Задание	5
Выполнение лабораторной работы	7
Задание 1. Рис. [-@fig:001].	7
Задание 2. Рис. [-@fig:002]-[-@fig:003].	8
Задание 3. Рис. [-@fig:004].	9
Выводы	11
Контрольные вопросы	12

Список иллюстраций

0.1	Название рисунка	7
0.2	1. Выполнение скрипта н.1	8
0.3	2. Выполнение скрипта н.2	9
0.4	3.Выполнение скрипта н.2	9
0.5	4.Выполнение скрипта н.3	10

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 имела 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

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. [-@fig:001])

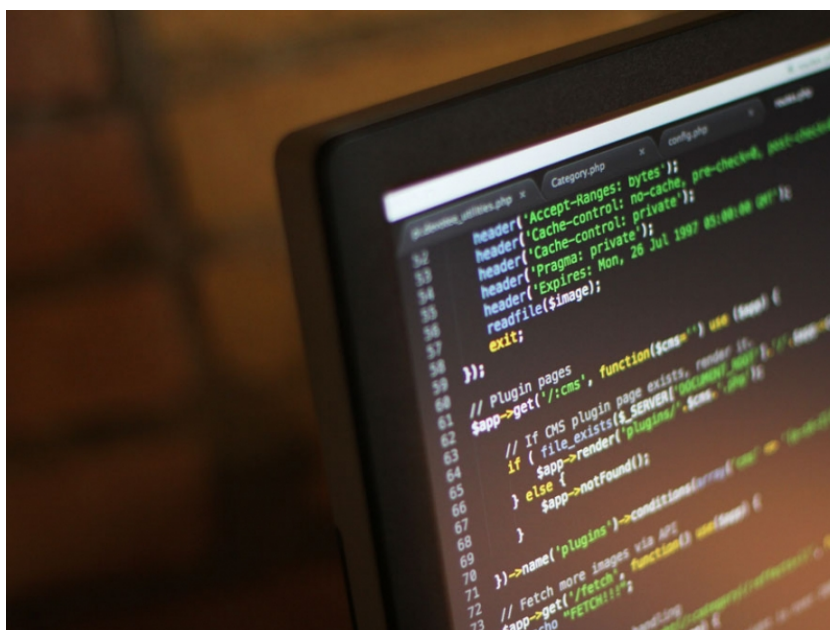


Рис. 0.1: Название рисунка

Задание 1. Рис. [-@fig:001].

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

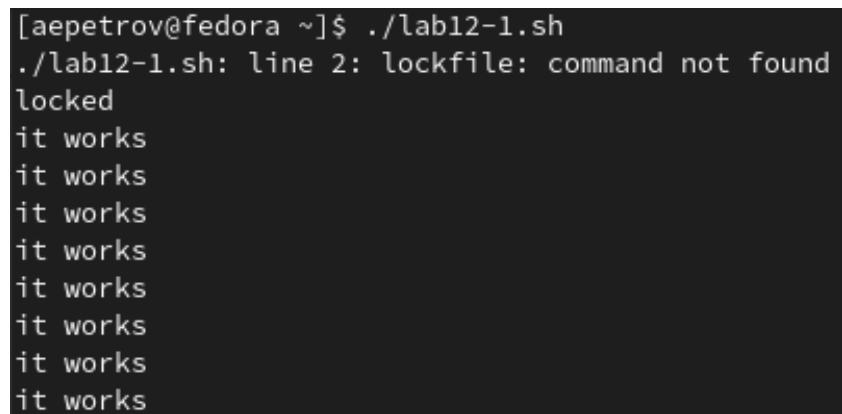
```
lockfile = "./lock.file"
```

```
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

```

Рис. 0.2: 1. Выполнение скрипта н.1

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
```

Рис. 0.3: 2. Выполнение скрипта н.2

```
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 1.27 (file 1 of 10)
```

Рис. 0.4: 3. Выполнение скрипта н.2

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
```

Рис. 0.5: 4.Выполнение скрипта н.3

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

Контрольные вопросы

1. Для правильного исполнения командной строки `while [$1 != "exit"]` квадратные скобки нужно поменять на круглые скобки
2. Использовать знак `$` между двумя переменными символьного типа для объединения нескольких строк в одно единую целую строку
3. `seq` - выводит последовательность целых или действительных чисел для передачи в другие программы, на языке `bash` её можно реализовать с помощью цикла `for`
4. $10 / 3 = 3$ (дробная часть будет отброшена, так как числа целые, но если бы было $10.000/3.000 = 3.333$)
5. `Zsh` выполнен на основе `bash`, а поэтому наследует как недостатки так и преимущества `bash`, но в `zsh` есть автодополнение, горячие клавиши, алиасы, различные удобства в виде тем, плагинов и расширенной поддержке, однако есть минус всей этой красоты и удобства, на узкоспециализированных машинах это красоты быть не может в принципе из-за ограничений железа или системного администратора
6. Синтаксис верен, однако для надежности лучше писать `for` и `((some words))` без пробела, т.е. слитно