

# Отчет к лабораторной работе №12

## Дисциплина: Операционные системы

Маслова Анастасия Сергеевна

### Содержание

#### Цель работы

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

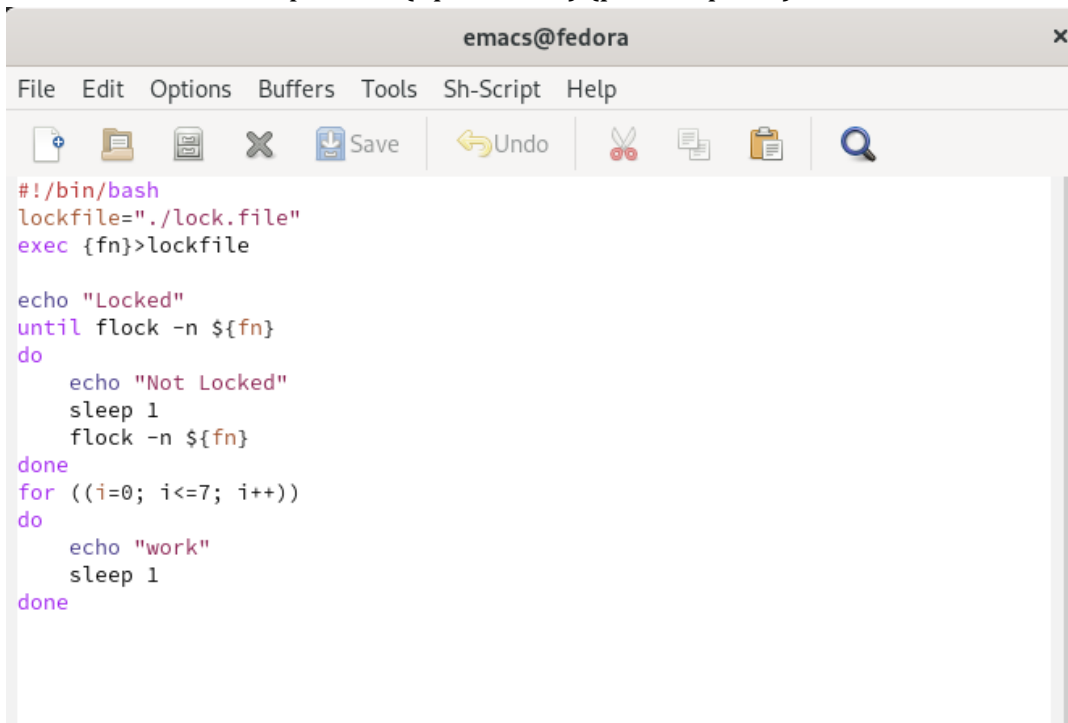
#### Задание

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

1. Я написала командный файл, реализующий упрощённый механизм семафоров. Данный командный файл в течение некоторого времени  $t_1$  дожидается

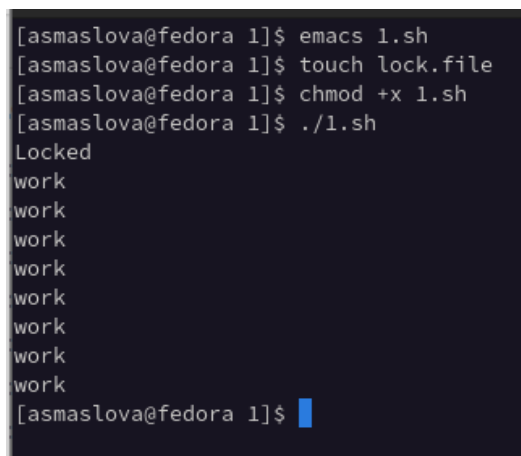
освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использует его в течение некоторого времени  $t_2 < t_1$ , также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом) (рис.1 и рис.2).



```
emacs@fedora
File Edit Options Buffers Tools Sh-Script Help
Save Undo
#!/bin/bash
lockfile="./lock.file"
exec {fn}>lockfile

echo "Locked"
until flock -n ${fn}
do
    echo "Not Locked"
    sleep 1
    flock -n ${fn}
done
for ((i=0; i<=7; i++))
do
    echo "work"
    sleep 1
done
```

рис.1 Текст программы



```
[asmaslova@fedora 1]$ emacs 1.sh
[asmaslova@fedora 1]$ touch lock.file
[asmaslova@fedora 1]$ chmod +x 1.sh
[asmaslova@fedora 1]$ ./1.sh
Locked
work
work
work
work
work
work
work
work
work
[asmaslova@fedora 1]$
```

рис.2 Работа программы

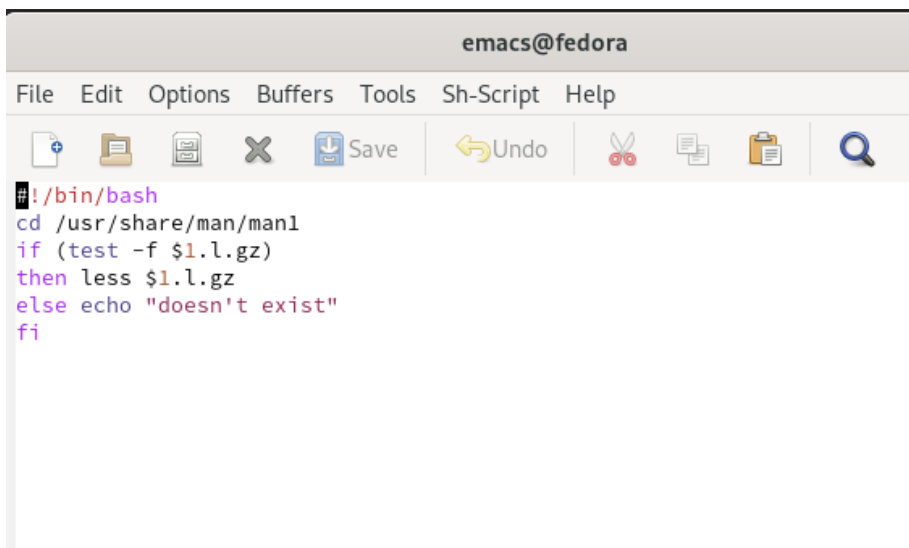
2. Далее я написала командный файл, который получает в виде аргумента командной строки название команды и в виде результата выдает справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге `man1`. Перед этим я изучила содержимое каталога `/usr/share/man/man1` (рис.3, рис.4 и рис.5).

```

[asmaslova@fedora ~]$ ls
bin  dev  home  lib64  media  opt  root  sbin  sys  usr
boot  etc  lib  lost+found  mnt  proc  run  srv  tmp  var
[asmaslova@fedora ~]$ cd /usr/share/man/man1
[asmaslova@fedora man1]$ ls
:.1.gz
'.1.gz'
ab.1.gz
abrt.1.gz
abrt-action-analyze-backtrace.1.gz
abrt-action-analyze-c.1.gz
abrt-action-analyze-ccpp-local.1.gz
abrt-action-analyze-core.1.gz
abrt-action-analyze-java.1.gz
abrt-action-analyze-oops.1.gz
abrt-action-analyze-python.1.gz
abrt-action-analyze-vmcore.1.gz
abrt-action-analyze-vulnerability.1.gz
abrt-action-analyze-xorg.1.gz
abrt-action-check-oops-for-hw-error.1.gz
abrt-action-find-bodhi-update.1.gz
abrt-action-generate-backtrace.1.gz
abrt-action-generate-core-backtrace.1.gz
abrt-action-install-debuginfo.1.gz
abrt-action-list-dsos.1.gz
abrt-action-notify.1.gz
abrt-action-perform-ccpp-analysis.1.gz
abrt-action-save-package-data.1.gz
abrt-action-trim-files.1.gz
abrt-applet.1.gz
abrt-auto-reporting.1.gz
abrt-bodhi.1.gz
abrt-cli.1.gz
abrt-dump-journal-core.1.gz
abrt-dump-journal-oops.1.gz
abrt-dump-journal-xorg.1.gz
abrt-dump-oops.1.gz
abrt-dump-xorg.1.gz
abrt-handle-upload.1.gz
abrt-harvest-pstoreoops.1.gz
abrt-harvest-vmcore.1.gz
abrt-merge-pstoreoops.1.gz
abrt-retrace-client.1.gz
abrt-server.1.gz
abrt-watch-log.1.gz
-- * --

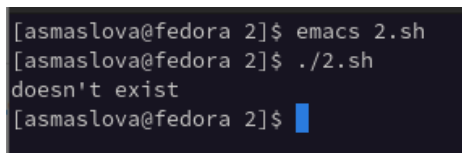
```

рис.3 Содержимое каталога /usr/share/man/man1



```
#!/bin/bash
cd /usr/share/man/man1
if (test -f $1.l.gz)
then less $1.l.gz
else echo "doesn't exist"
fi
```

рис.4 Текст программы



```
[asmaslova@fedora 2]$ emacs 2.sh
[asmaslova@fedora 2]$ ./2.sh
doesn't exist
[asmaslova@fedora 2]$
```

рис.5 Работа программы

- Используя встроенную переменную \$RANDOM, я написала командный файл, генерирующий случайную последовательность букв латинского алфавита (рис.6 и рис.7)

```
for symbol in {A..Z} {a..z}
do SYMBOLS=$SYMBOLS$symbol;
done
STR_LEN=20
STR=""
for i in seq 1 $STR_LEN
do
    STR=$STR${SYMBOLS:${(expr $RANDOM % ${#SYMBOLS}):1}}
done
echo $STR
```

рис.6 Текст программы

```
[asmaslova@fedora 3]$ ./3.sh
giP
[asmaslova@fedora 3]$ ./3.sh
Gdq
[asmaslova@fedora 3]$ ./3.sh
yNk
[asmaslova@fedora 3]$ ./3.sh
hZX
[asmaslova@fedora 3]$ ./3.sh
UfF
[asmaslova@fedora 3]$ ./3.sh
ELN
[asmaslova@fedora 3]$ ./3.sh
woD
[asmaslova@fedora 3]$ ./3.sh
yKp
[asmaslova@fedora 3]$ ./3.sh
Oqf
```

*рис.7 Работа программы*

## Выводы

В ходе лабораторной работы я изучила основы программирования в оболочке ОС UNIX и научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## Список литературы