

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №12**

дисциплина: Операционные системы

**Студент:** Султанова Лейла

**Группа:** НБИбд-03-22

Москва

2023 г.

## Цель работы

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

## Последовательность выполнения работы

1. Напишем 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). Запустим 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. Доработаем программу так, чтобы имелась возможность взаимодействия трёх и более процессов.

```
[lsultanova@lsultanova ~]$ touch lab12.sh
[lsultanova@lsultanova ~]$ chmod +x lab12.sh
```

```
lab12.sh [ -M-- ] 11 L [ 1+ 7 8/ 15] *(122 / 288b) 0010 0x00A
#!/bin/bash
lockfile="./lockfile"
exec {fn}>$lockfile
echo "lock"
until flock -n ${fn}
do
    echo "not lock"
    sleep 1
    flock -n ${fn}
done
for ((i=0;i<=5; i++))
do
    echo "work"
    sleep 1
done
```

```
lock
work
work
work
work
work
work
```

2. Реализуем команду man с помощью командного файла. Изучим 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.

```
lsultanova@lsultanova ~]$ touch lab12-1.sh
lsultanova@lsultanova ~]$ chmod lab12-1.sh
chmod: пропущен операнд после «lab12-1.sh»
по команде «chmod --help» можно получить дополнительную информацию.
lsultanova@lsultanova ~]$ chmod +x lab12-1.sh
lsultanova@lsultanova ~]$

lab12-1.sh      [-M-]  8 L: [ 1+ 2 3/ 3] *(43 / 43b) <EO>
#!/bin/bash
cd /usr/share/man/man1
less $1

BASH_BUILTINS(1)                                General Commands Manual                                BASH_BUILTINS(1)

ESC[1mNAMEESC[0m
., ., [, alias, bg, bind, break, builtin, caller, cd, command, compgen, complete, compopt, continue, declare, dirs, disown, echo, enable, eval, exec, exit, export, false,
fc, fg, getopts, hash, help, history, jobs, kill, let, local, logout, mapfile, popd, printf, pushd, pwd, read, readarray, readonly, return, set, shift, shopt, source,
suspend, test, times, trap, true, type, typeset, ulimit, umask, unalias, unset, wait - bash built-in commands, see ESC[1mbashESC[22m(1)

ESC[1mBASH BUILTIN COMMANDSESC[0m
Unless otherwise noted, each builtin command documented in this section as accepting options preceded by ESC[1m- ESC[22maccepts ESC[1m-- ESC[22mto signify the end of the o
ptions. The ESC[1m-ESC[22m, ESC[1mtrueESC[22m,
ESC[1mfalseESC[22m, and ESC[1mtestESC[22mESC[1m[ ESC[22mbuiltins do not accept options and do not treat ESC[1m-- ESC[22mspecially. The ESC[1mexitESC[22m, ESC[1mlogoutESC[22m
, ESC[1mreturnESC[22m, ESC[1mbreakESC[22m, ESC[1mcontinueESC[22m, ESC[1mletESC[22m, and ESC[1mshift ESC[22mbuiltins accept and process argu
ments beginning with ESC[1m- ESC[22mwithout requiring ESC[1m-ESC[22m. Other builtins that accept arguments but are not specified as accepting options interpret arguments
s beginning with ESC[1m- ESC[22mas
invalid options and require ESC[1m-- ESC[22mto prevent this interpretation.
ESC[1m- ESC[22mESC[4margumentsESC[24m
No effect; the command does nothing beyond expanding ESC[4margumentsESC[24m and performing any specified redirections. The return status is zero.

ESC[1m- ESC[4mESC[22mfilenameESC[24m ESC[4margumentsESC[24m]
ESC[1msource ESC[4mESC[22mfilenameESC[24m ESC[4margumentsESC[24m]
Read and execute commands from ESC[4mfilenameESC[24m in the current shell environment and return the exit status of the last command executed from ESC[4mfilenameESC[24m
. If ESC[4mfilenameESC[24m does
not contain a slash, filenames in ESC[1mPATH ESC[22mare used to find the directory containing ESC[4mfilenameESC[24m, but ESC[4mfilenameESC[24m does not need to be ex
cutable. The file searched for in
ESC[1mPATH ESC[22mneed not be executable. When ESC[1mbash ESC[22mis not in ESC[4mposixESC[24m ESC[4mmodeESC[24m, it searches the current directory if no file is found
in ESC[1mPATH ESC[22m. If the ESC[1msourcepath ESC[22moption to the ESC[1mshoptESC[0m
builtin command is turned off, the ESC[1mPATH ESC[22mis not searched. If any ESC[4margumentsESC[24m are supplied, they become the positional parameters when ESC[4mfi
lenameESC[24m is executed. Other-
wise the positional parameters are unchanged. If the ESC[1m-T ESC[22moption is enabled, ESC[1m- ESC[22minherits any trap on ESC[1mDEBUGESC[22m; if it is not, any ESC[1
mDEBUG ESC[22mtrap string is saved and restored
around the call to ESC[1m- ESC[22m, and ESC[1m- ESC[22munsets the ESC[1mDEBUG ESC[22mtrap while it executes. If ESC[1m-T ESC[22mis not set, and the sourced file changes
the ESC[1mDEBUG ESC[22mtrap, the new value is retained when
ESC[1m- ESC[22mcompletes. The return status is the status of the last command exited within the script (0 if no commands are executed), and false if ESC[4mfilenameESC[24
```

3. Используя встроенную переменную \$RANDOM, напомним командный файл, генерирующий случайную последовательность букв латинского алфавита. Учитываем, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

```
lsultanova@lsultanova ~]$ ./lab12-1.sh
lsultanova@lsultanova ~]$ touch lab12-2.sh
lsultanova@lsultanova ~]$ chmod +x lab12-2.sh
lsultanova@lsultanova ~]$
```

```

lab12-2.sh [----] 5 L: [ 1+ 6 7/ 15] *(58 / 219b) 0032 0x020
#!/bin/bash
M=10
c=1
d=1
echo
echo "10 random words"
while [ ($c!=($M+1)) ]
do
    echo $(for((i=1;i<=10;i++)); do printf '%s' "${RANDOM:0:1}"; done) | tr '0-9' 'a-z'
    echo $d
    ((c+=1))
    ((d+=1))
done

```

```

10 random words
begibccbcd
1
cfebccddb
2
bbicbbdccc
3
ccjcbcbbeb
4
cdbc bcddbb
5
ccjibccddb
6
tebccbcbdd
7
fcbbgejbb
8
bibbbhfcde
9
jbgbdhcedc
10

```