

Лабораторная работа № 14

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Мальянц В. К.

10 мая 2025

Российский университет дружбы народов, Москва, Россия

Цель работы

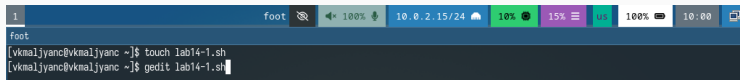
- Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Задание

- Задание № 1. Написать командный файл, реализующий упрощённый механизм семафоров.
- Задание № 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`.
- Задание № 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

- Создаю файл lab14-1.sh и открываю его (рис. 1).



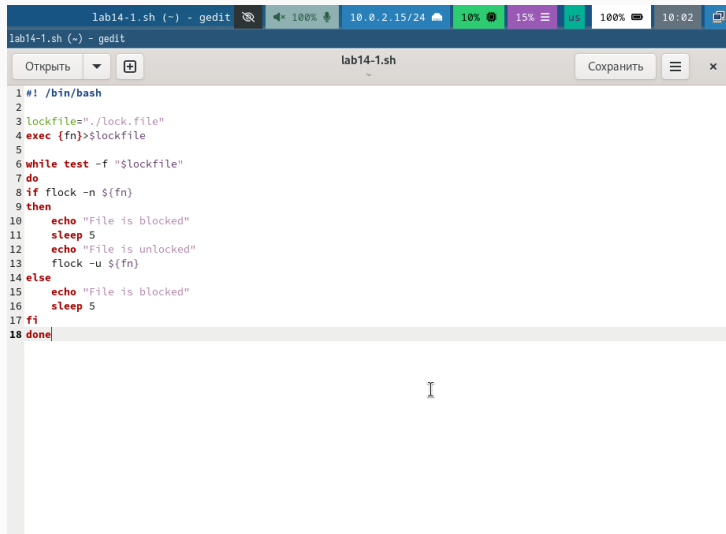
The screenshot shows a terminal window with a dark blue header bar. The header bar contains the text 'foot' on the left, and a series of status icons and text on the right: a network icon, '100%', '10.0.2.15/24', '10%', '15%', '100%', and '10:00'. The terminal content shows the prompt '[vkmajyanc@vkmajyanc ~]' followed by the command 'touch lab14-1.sh' and the command 'gedit lab14-1.sh' with a cursor at the end of the second line.

```
foot
[vkmajyanc@vkmajyanc ~]$ touch lab14-1.sh
[vkmajyanc@vkmajyanc ~]$ gedit lab14-1.sh
```

Рис. 1: Создание файла lab14-1.sh и открытие его


Задание № 1

- Ввожу код в файл lab14-1.sh (рис. 2).



```
lab14-1.sh (~) - gedit
lab14-1.sh (~) - gedit
Открыть  + lab14-1.sh Сохранить  ×
1 #!/bin/bash
2
3 lockfile='./lock.file'
4 exec {fn}>$lockfile
5
6 while test -f "$lockfile"
7 do
8   if flock -n ${fn}
9   then
10    echo "File is blocked"
11    sleep 5
12    echo "File is unlocked"
13    flock -u ${fn}
14   else
15    echo "File is blocked"
16    sleep 5
17   fi
18 done
```


- Даю право на исполнение файла lab14-1.sh и запускаю его. Убеждаюсь в том, что программа работает корректно (рис. 3).

A terminal window with a dark background and a colorful abstract pattern on the right. The terminal text shows a user setting permissions and running a script that prints alternating 'File is blocked' and 'File is unlocked' messages.

```
[vkmajyanc@vkmajyanc ~]$ chmod +x lab14-1.sh
[vkmajyanc@vkmajyanc ~]$ bash lab14-1.sh
File is blocked
File is unlocked
File is blocked
File is unlocked
File is blocked
File is unlocked
File is blocked
File is unlocked
File is blocked
File is unlocked
File is blocked
File is unlocked
File is blocked
```


Рис. 3: Право на исполнение файла lab14-1.sh и запуск этого файла

Задание № 2

- Просматриваю содержимое /usr/share/man/man1 (рис. 4).

```
[vkmajjanc@vkmajjanc ~]$ ls /usr/share/man/man1
.:1.gz
'[:1.gz'
7z.1.gz
a2ping.1.gz
ac.1.gz
aconect.1.gz
addr2line.1.gz
afm2afm.1.gz
afm2pl.1.gz
afm2tfm.1.gz
albatross.1.gz
aleph.1.gz
alias.1.gz
allcm.1.gz
allec.1.gz
allneeded.1.gz
alsactl.1.gz
alsa-info.sh.1.gz
alsaloop.1.gz
alsamixer.1.gz
alsamute.1.gz
alt-java.1.gz
alt-java-java-21-openjdk.1.gz
amidi.1.gz
amixer.1.gz
amstex.1.gz
animate.1.gz
aplay.1.gz
aplaymidi.1.gz
aplaymidi2.1.gz
appstreamcli.1.gz
apropos.1.gz
git-write-tree.1.gz
glib-compile-schemas.1.gz
gn.1.gz
gnake.1.gz
gneqn.1.gz
gnome-keyring.1.gz
gnome-keyring-3.1.gz
gnome-keyring-daemon.1.gz
gnroff.1.gz
goid-tool.1.gz
gp-archive.1.gz
gpaswd.1.gz
gp-collect-app.1.gz
gp-display-html.1.gz
gp-display-src.1.gz
gp-display-text.1.gz
gpg.1.gz
gpg2.1.gz
gpg-agent.1.gz
gpg-card.1.gz
gpg-check-pattern.1.gz
gpgconf.1.gz
gpg-connect-agent.1.gz
gpgparsemail.1.gz
gpg-preset-passphrase.1.gz
gpgsm.1.gz
gpgtar.1.gz
gpgv.1.gz
gpgv2.1.gz
gpg-wks-client.1.gz
gpg-wks-server.1.gz
gpic.1.gz
openssl-x509.1.gz
opentv.1.gz
opl2ofm.1.gz
optex.1.gz
ot2kpx.1.gz
otangle.1.gz
otfinfo.1.gz
otftotfm.1.gz
otp2ocp.1.gz
outocp.1.gz
ovf2ovp.1.gz
ovp2ovf.1.gz
pacat.1.gz
pacmd.1.gz
pactl.1.gz
padsp.1.gz
pagelayoutapi.1.gz
panon.1.gz
panphletangler.1.gz
panelctl.1.gz
pango-view.1.gz
paper.1.gz
paperconf.1.gz
paplay.1.gz
paps.1.gz
parec.1.gz
parecord.1.gz
passwd.1.gz
passwd.1.gz
paste.1.gz
pasuspender.1.gz
patch.1.gz
```

- Создаю файл lab14-2.sh и открываю его (рис. 5).

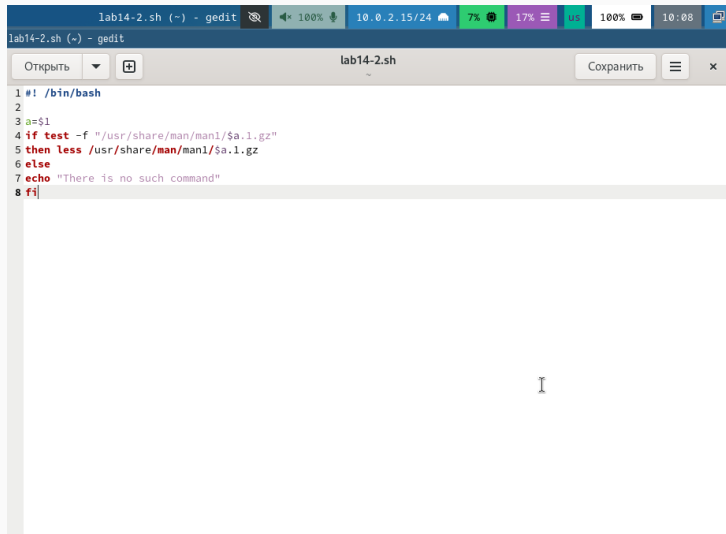
A terminal window with a dark background. The prompt is [vkmajyanc@vkmajyanc ~]. The first command is touch lab14-2.sh. The second command is gedit lab14-2.sh, followed by a cursor.

```
[vkmajyanc@vkmajyanc ~]$ touch lab14-2.sh  
[vkmajyanc@vkmajyanc ~]$ gedit lab14-2.sh
```

Рис. 5: Создание файла lab14-2.sh и открытие его


Задание № 2

- Ввожу код в файл lab14-2.sh (рис. 6).



```
lab14-2.sh (~) - gedit
lab14-2.sh (~) - gedit
Открыть  +  lab14-2.sh  Сохранить  ×
1  #!/bin/bash
2
3  a=$1
4  if test -f "/usr/share/man/man1/$a.1.gz"
5  then less /usr/share/man/man1/$a.1.gz
6  else
7  echo "There is no such command"
8  fi
```

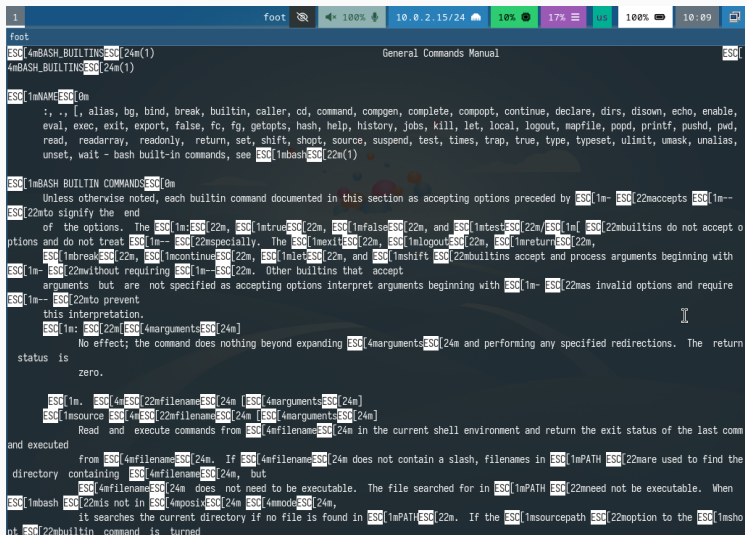
- Даю право на исполнение файла lab14-2.sh и запускаю его (рис. 7).



```
[vkmajyanc@vkmajyanc ~]$ chmod +x lab14-2.sh  
[vkmajyanc@vkmajyanc ~]$ bash lab14-2.sh cd
```

Рис. 7: Право на исполнение файла lab14-2.sh и запуск этого файла

- Убеждаюсь в том, что программа работает корректно (рис. 8).




```
1 foot 10.0.2.15/24 10% 17% us 100% 10:09
foot
ESC[4mBASH_BUILTINESC[24m(1) General Commands Manual ESC[
4mBASH_BUILTINESC[24m(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 options. The ESC[1m:ESC[22m, ESC[1mtrueESC[22m, ESC[1mfalseESC[22m, and ESC[1mtestESC[22m/ESC[1mESC[22mbuiltins do not accept o
ptions 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 arguments 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 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 comm
and 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 executable. The file searched for in ESC[1mPATH ESC[22mneed not be executable. When
ESC[1mbash ESC[22mis not in ESC[4mposixESC[24m ESC[4mmodESC[24m,
it searches the current directory if no file is found in ESC[1mPATHESC[22m. If the ESC[1msourcepath ESC[22moption to the ESC[1msho
ot ESC[22mbuiltin command is turned
```

- Создаю файл lab14-3.sh и открываю его (рис. 9).

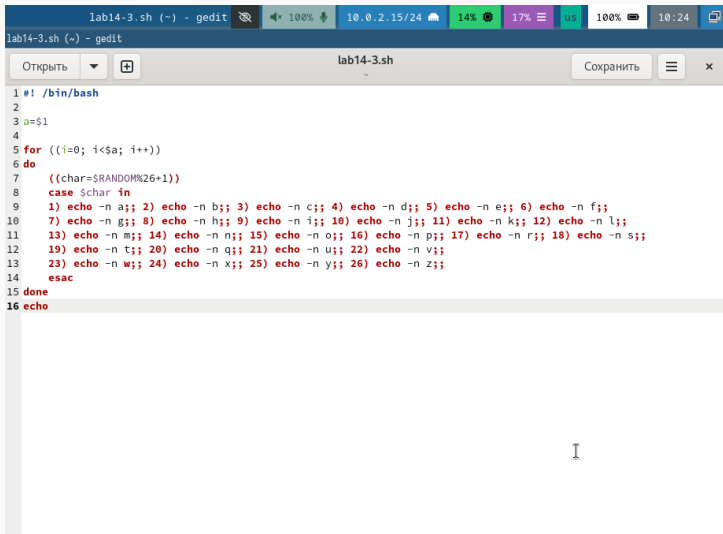
A terminal window with a dark background. The prompt is [vkmajyanc@vkmajyanc ~]. The first command is touch lab14-3.sh. The second command is gedit lab14-3.sh, followed by a cursor.

```
[vkmajyanc@vkmajyanc ~]$ touch lab14-3.sh  
[vkmajyanc@vkmajyanc ~]$ gedit lab14-3.sh
```

Рис. 9: Создание файла lab14-3.sh и открытие его

Задание № 3

- Ввожу код в файл lab14-3.sh (рис. 10).



```
lab14-3.sh (~) - gedit
lab14-3.sh (~) - gedit
Открыть  lab14-3.sh  Сохранить  x

1 #! /bin/bash
2
3 a=$1
4
5 for ((i=0; i<$a; i++))
6 do
7     ((char=$RANDOM%26+1))
8     case $char in
9         1) echo -n a;; 2) echo -n b;; 3) echo -n c;; 4) echo -n d;; 5) echo -n e;; 6) echo -n f;;
10        7) echo -n g;; 8) echo -n h;; 9) echo -n i;; 10) echo -n j;; 11) echo -n k;; 12) echo -n l;;
11        13) echo -n m;; 14) echo -n n;; 15) echo -n o;; 16) echo -n p;; 17) echo -n r;; 18) echo -n s;;
12        19) echo -n t;; 20) echo -n q;; 21) echo -n u;; 22) echo -n v;;
13        23) echo -n w;; 24) echo -n x;; 25) echo -n y;; 26) echo -n z;;
14    esac
15 done
16 echo
```


- Даю право на исполнение файла lab14-3.sh и запускаю его. Убеждаюсь в том, что программа работает корректно (рис. 11).

```
[vkmajyanc@vkmajyanc ~]$ chmod +x lab14-3.sh  
[vkmajyanc@vkmajyanc ~]$ bash lab14-3.sh 15  
uiojaipzxuujlpb  
[vkmajyanc@vkmajyanc ~]$
```

Рис. 11: Право на исполнение файла lab14-3.sh и запуск этого файла

Выводы

- Я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Спасибо за внимание
