

Программирование в командном процессоре ОС UNIX.

Исаева Зарина¹

20 апреля, 2024, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

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

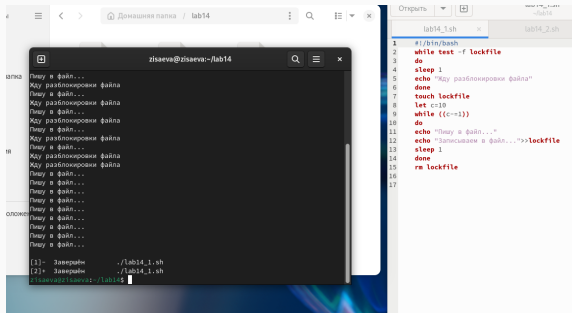
Задачи лабораторной работы

1 Выполнить 3 задания

Процесс выполнения лабораторной работы

1. Написали командный файл, реализующий упрощённый механизм семафоров. 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).

Выполнение работы



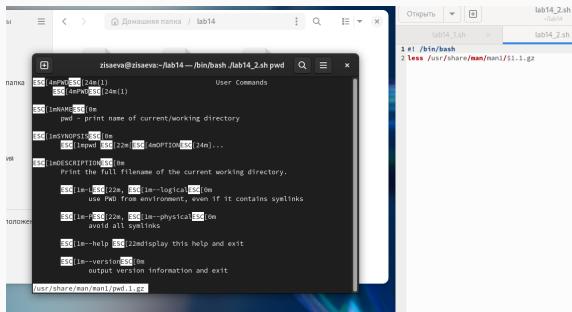
The screenshot displays a Linux desktop environment. On the left, a terminal window titled 'zisaeva@zisaeva: ~/lab14' shows the execution of a script. The output consists of a series of messages: 'Пишу в файл...', 'Жду разблокировки файла', and 'Пишу в файл...', which are repeated multiple times. At the bottom of the terminal, two prompts are visible: '[1]- Завершаем ./lab14_1.sh' and '[2]+ Завершаем ./lab14_1.sh'. On the right, a file manager window shows the contents of the 'lab14' directory. It contains two files: 'lab14_1.sh' and 'lab14_2.sh'. The 'lab14_1.sh' file is selected, and its contents are displayed in a text editor. The script is a shell script that implements a file locking mechanism. It starts with a shebang line, followed by a while loop that checks if a file named 'lockfile' exists. If it does, the script sleeps for 1 second and echoes a message. Once the file is removed, it touches the lockfile, sets a counter 'c' to 10, and enters another while loop that decrements 'c' until it reaches 0. During this second loop, it echoes a message and appends the current time to the lockfile. Finally, it removes the lockfile.

```
1 #!/bin/bash
2 while test -f lockfile
3 do
4     sleep 1
5     echo "Жду разблокировки файла"
6 done
7 touch lockfile
8 let c=10
9 while ((c>=1))
10 do
11     echo "Пишу в файл..."
12     echo "$(date +%Y-%m-%d %H:%M:%S)">>lockfile
13     sleep 1
14 done
15 rm lockfile
16
17
```

Рис. 1: Задание 1

2. Реализовали команду `man` с помощью командного файла. Изучили содержимое каталога `/usr/share/man/man1`. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд.

Выполнение работы



The image shows a desktop environment with a file manager and a terminal window. The file manager is open to the directory `/lab14`. It shows two files: `lab14_1.sh` and `lab14_2.sh`. The `lab14_2.sh` file is selected, and its contents are visible in the right pane:

```
1 #! /bin/bash
2 less /usr/share/man/man1/pwd.1.gz
```

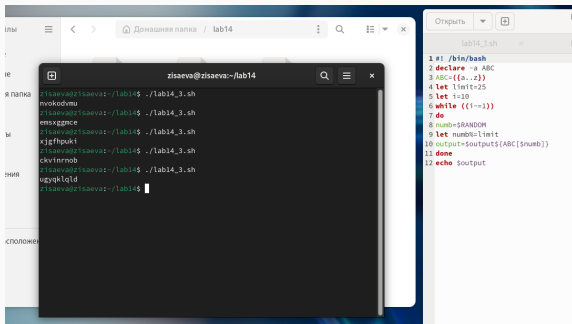
The terminal window is titled `zisaeva@zisaeva:~/lab14 — /bin/bash /lab14_2.sh pwd`. It shows the execution of the `pwd` command, which prints the full filename of the current working directory. The output is `/usr/share/man/man1/pwd.1.gz`.

```
zisaeva@zisaeva:~/lab14 — /bin/bash /lab14_2.sh pwd
ESC[4mPwdESC[24m(1) User Commands
ESC[1mNAMEESC[0m
pwd — print name of current/working directory
ESC[1mSYNOPSISESC[0m
ESC[4m(1)pwd ESC[22m[ESC[4mOPTIONESC[24m]...
ESC[1mDESCRIPTIONESC[0m
Print the full filename of the current working directory.
ESC[4m(1)ESC[22m, ESC[4m(1)ESC[22m[ESC[4mOPTIONESC[24m]
use PWD from environment, even if it contains symlinks
ESC[4m(1)ESC[22m, ESC[4m(1)ESC[22m[ESC[4mOPTIONESC[24m]
avoid all symlinks
ESC[4m(1)ESC[22m[ESC[4mOPTIONESC[24m]
display this help and exit
ESC[4m(1)ESC[22m[ESC[4mOPTIONESC[24m]
output version information and exit
/usr/share/man/man1/pwd.1.gz
```

Рис. 2: Задание 2

3. Используя встроенную переменную \$RANDOM , написали командный файл, генерирующий случайную последовательность букв латинского алфавита

Выполнение работы



The image shows a desktop environment with a terminal window and a file editor. The terminal window, titled 'zisaeva@zisaeva: ~/lab14', displays the execution of a script named 'lab14_3.sh'. The script's output consists of several lines of random characters: 'nvokodvnu', 'emsggcce', 'xjgfhpuKt', 'ckvlnrnob', and 'ugyqklqld'. The file editor, titled 'lab14_1.sh', shows the source code of the script. The code includes a shebang, variable declarations, a loop, and a function call.

```
zisaeva@zisaeva: ~/lab14$ ./lab14_3.sh
nvokodvnu
zisaeva@zisaeva: ~/lab14$ ./lab14_3.sh
emsggcce
zisaeva@zisaeva: ~/lab14$ ./lab14_3.sh
xjgfhpuKt
zisaeva@zisaeva: ~/lab14$ ./lab14_3.sh
ckvlnrnob
zisaeva@zisaeva: ~/lab14$ ./lab14_3.sh
ugyqklqld
zisaeva@zisaeva: ~/lab14$
```

```
1 #! /bin/bash
2 declare -a ABC
3 ABC=({a..z})
4 let limit=25
5 let i=10
6 while ((i--))
7 do
8   numb=$((RANDOM))
9   let numb%=limit
10  output=$(outpu${ABC[$numb])
11 done
12 echo $output
```

Рис. 3: Задание 3

Выводы по проделанной работе

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