

Программирование в командном процессоре ОС UNIX.

Марьям Ел Вакил¹

17 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

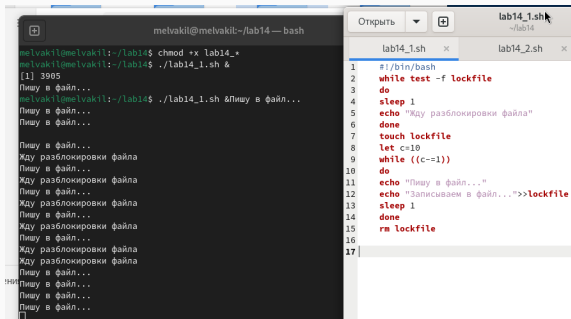
Задачи лабораторной работы

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 image shows a terminal window on the left and a script file on the right. The terminal window is titled 'melvakil@melvakil:~/lab14 — bash'. It shows the following commands and output:

```
melvakil@melvakil:~/lab14$ chmod +x lab14.*
melvakil@melvakil:~/lab14$ ./lab14_1.sh &
[1] 3905
Пишу в файл...
melvakil@melvakil:~/lab14$ ./lab14_1.sh &Пишу в файл...
Пишу в файл...
Пишу в файл...

Пишу в файл...
Жду разблокировки файла
Пишу в файл...
Жду разблокировки файла
Пишу в файл...
Жду разблокировки файла
Пишу в файл...
Жду разблокировки файла
Пишу в файл...
Жду разблокировки файла
Жду разблокировки файла
Пишу в файл...
Пишу в файл...
Пишу в файл...
Пишу в файл...
```

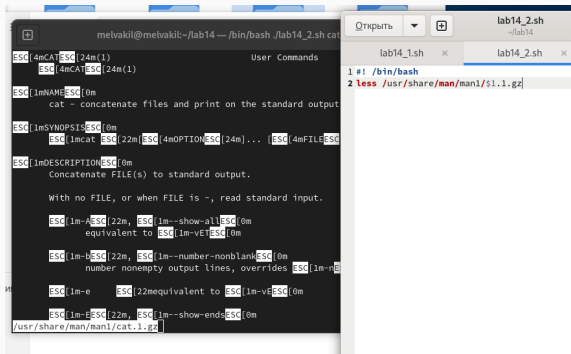
The script file on the right is titled 'lab14_1.sh' and contains the following code:

```
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 "Записываем в файл...">>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 two overlapping windows. The background window is a terminal titled 'melvakil@melvakil:~/lab14 — /bin/bash ./lab14_2.sh cat'. It displays the output of a script that explains the 'cat' command. The output includes the command 'cat - concatenate files and print on the standard output', its synopsis, and a description. It also shows how to use 'cat' with flags like '-a' (show all) and '-n' (number nonempty output lines). The foreground window is a file manager titled 'lab14_2.sh' showing the contents of the script file. The script has two lines: '1 #! /bin/bash' and '2 less /usr/share/man/man1/\$1.1.gz'.

```
melvakil@melvakil:~/lab14 — /bin/bash ./lab14_2.sh cat
ESC[4mCATESC[24m(1)                                User Commands
ESC[4mCATESC[24m(1)
ESC[1mNAMEESC[0m
cat - concatenate files and print on the standard output
ESC[1mSYNOPSISESC[0m
ESC[1mcat ESC[22m[ESC[4mOPTIONESC[24m]... ESC[4mFILEESC[0m
ESC[1mDESCRIPTIONESC[0m
Concatenate FILE(s) to standard output.

With no FILE, or when FILE is -, read standard input.

ESC[1m-ASESC[22m, ESC[1m--show-allESC[0m
equivalent to ESC[1m-vESC[0m

ESC[1m-bESC[22m, ESC[1m--number-nonblankESC[0m
number nonempty output lines, overrides ESC[1m-nESC[0m

ESC[1m-eESC[22mequivalent to ESC[1m-vESC[0m

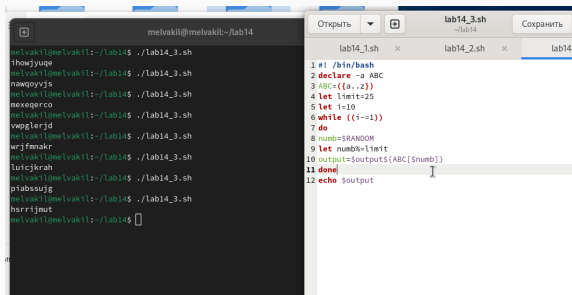
ESC[1m-ESCE[22m, ESC[1m--show-endESC[0m
/usr/share/man/man1/cat.1.gz

Открыть  lab14_2.sh
lab14_1.sh x lab14_2.sh x
1 #! /bin/bash
2 less /usr/share/man/man1/$1.1.gz
```

Рис. 2: Задание 2

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

Выполнение работы



The image shows two terminal windows. The left window, titled 'melvakil@melvakil:~/lab14', shows a series of commands being executed in a loop, each followed by the output of a script named 'lab14_3.sh'. The commands are: '1howjyuge', '2elvak10melvakil:~/lab14\$./lab14_3.sh', '3awqoyvj3', '4elvak10melvakil:~/lab14\$./lab14_3.sh', '5exeqerco', '6elvak10melvakil:~/lab14\$./lab14_3.sh', '7vwpqlerjd', '8elvak10melvakil:~/lab14\$./lab14_3.sh', '9wrjfnakr', '10elvak10melvakil:~/lab14\$./lab14_3.sh', '11lutcjkrah', '12elvak10melvakil:~/lab14\$./lab14_3.sh', '13piabssujg', '14elvak10melvakil:~/lab14\$./lab14_3.sh', '15hsrrtjmut', and '16elvak10melvakil:~/lab14\$'. The right window, titled 'lab14_3.sh', shows the content of the script being executed. The script is a bash script that declares a variable 'ABC' with the value 'a..z', sets a 'limit' to 25, and enters a 'while' loop that runs 25 times. In each iteration, it generates a random number 'numb' between 1 and the limit, and appends 'ABC[numb]' to the 'output' variable. The script ends with 'done' and 'echo \$output'.

```
melvakil@melvakil:~/lab14$ ./lab14_3.sh
1howjyuge
melvakil@melvakil:~/lab14$ ./lab14_3.sh
2elvak10melvakil:~/lab14$ ./lab14_3.sh
3awqoyvj3
melvakil@melvakil:~/lab14$ ./lab14_3.sh
4elvak10melvakil:~/lab14$ ./lab14_3.sh
5exeqerco
melvakil@melvakil:~/lab14$ ./lab14_3.sh
6elvak10melvakil:~/lab14$ ./lab14_3.sh
7vwpqlerjd
melvakil@melvakil:~/lab14$ ./lab14_3.sh
8elvak10melvakil:~/lab14$ ./lab14_3.sh
9wrjfnakr
melvakil@melvakil:~/lab14$ ./lab14_3.sh
10elvak10melvakil:~/lab14$ ./lab14_3.sh
11lutcjkrah
melvakil@melvakil:~/lab14$ ./lab14_3.sh
12elvak10melvakil:~/lab14$ ./lab14_3.sh
13piabssujg
melvakil@melvakil:~/lab14$ ./lab14_3.sh
14elvak10melvakil:~/lab14$ ./lab14_3.sh
15hsrrtjmut
melvakil@melvakil:~/lab14$
```

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