

Операционные системы

Программирование в командном процессоре ОС UNIX.

Кайнова Екатерина Андреевна

27 апреля 2025

Российский университет дружбы народов, Москва, Россия

Цели и задачи работы

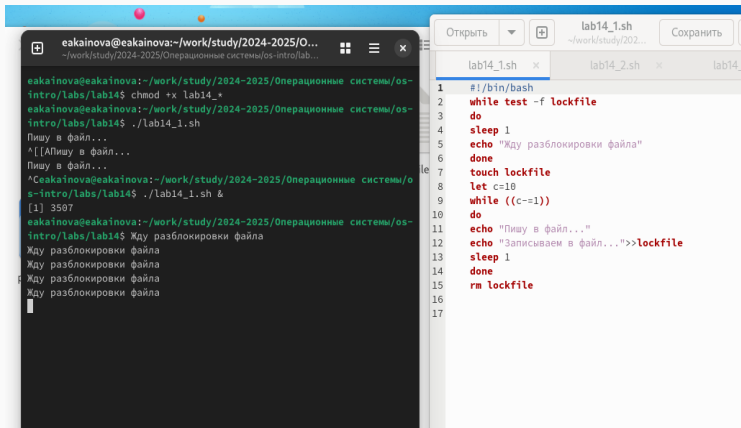
Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов

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 file editor on the right. The terminal window displays the execution of a script that attempts to write to a file, but it gets stuck in a loop because the file is locked. The file editor shows the script code, which uses a while loop to wait for the lock to be removed.

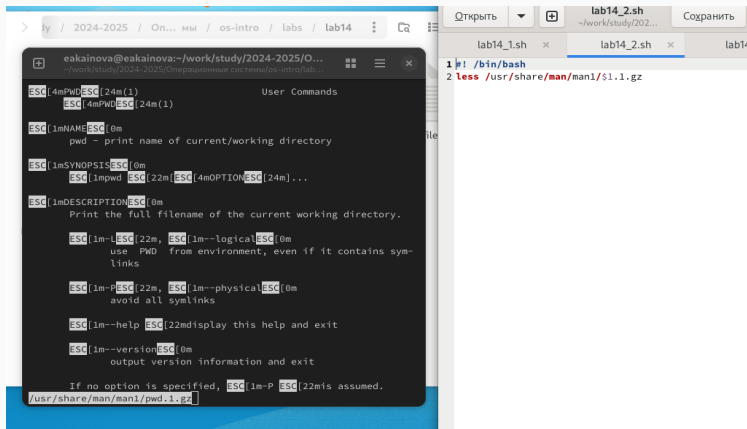
```
eakainova@eakainova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ chmod +x lab14_*
eakainova@eakainova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_1.sh
Пишу в файл...
^[[АПишу в файл...
Пишу в файл...
^Ceakainova@eakainova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_1.sh &
[1] 3507
eakainova@eakainova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
```

```
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



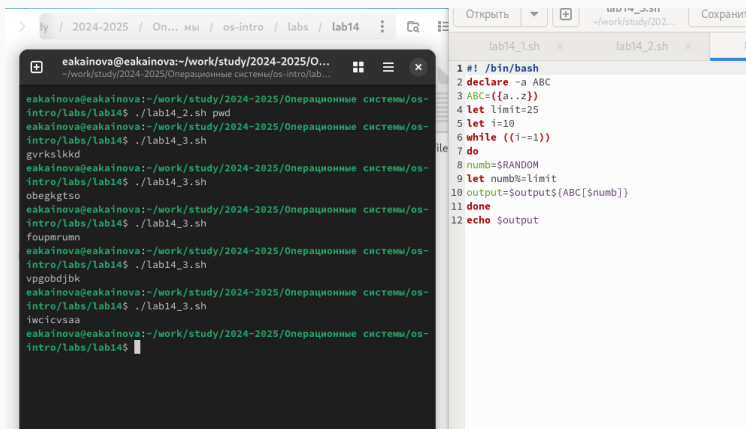
The image shows a terminal window and a web browser side-by-side. The terminal window, titled 'eakainova@eakainova:~/work/study/2024-2025/O...', displays a list of user commands and their descriptions. The web browser, titled 'lab14_2.sh', shows the content of a file named 'lab14_2.sh' which contains two lines of code: '1 #! /bin/bash' and '2 less /usr/share/man/man1/pwd.1.gz'.

```
eakainova@eakainova:~/work/study/2024-2025/O...  
~/work/study/2024-2025/Операционные системы/os-intro/lab...  
ESC[4mPWDESC[24m(1) User Commands  
ESC[4mPWDESC[24m(1)  
ESC[1mNAMEESC[0m  
pwd - print name of current/working directory  
ESC[1mSYNOPSISESC[0m  
ESC[1mpwd ESC[22mESC[4mOPTIONESC[24m...  
ESC[1mDESCRIPTIONESC[0m  
Print the full filename of the current working directory.  
ESC[1m-LESC[22m, ESC[1m--logicalESC[0m  
use PWD from environment, even if it contains sym-  
links  
ESC[1m-PESC[22m, ESC[1m--physicalESC[0m  
avoid all symlinks  
ESC[1m--help ESC[22mdisplay this help and exit  
ESC[1m--versionESC[0m  
output version information and exit  
If no option is specified, ESC(1m-P ESC(22mis assumed.  
/usr/share/man/man1/pwd.1.gz
```

```
lab14_2.sh  
1 #! /bin/bash  
2 less /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



The image shows a terminal window on the left and a code editor on the right. The terminal window displays the execution of a shell script named `lab14_3.sh` in a directory `~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14`. The script's output consists of random strings: `gvrkslkdd`, `obegkgts`, `foupmrumn`, `vpgobdjbk`, and `iwcicvsaa`. The code editor on the right shows the content of `lab14_1.sh`, which is a shell script that declares a variable `ABC`, sets a `limit` of 25, and uses a `while` loop to generate random output.

```
eakainova@eakainova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_2.sh pwd
eakainova@eakainova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
gvrkslkdd
eakainova@eakainova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
obegkgts
eakainova@eakainova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
foupmrumn
eakainova@eakainova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
vpgobdjbk
eakainova@eakainova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
iwcicvsaa
eakainova@eakainova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$
```

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