

Операционные системы

Программирование в командном процессоре ОС UNIX.

Оксана Чумаченко

2 сентября 2025

Российский университет дружбы народов, Москва, Россия

Цели и задачи работы

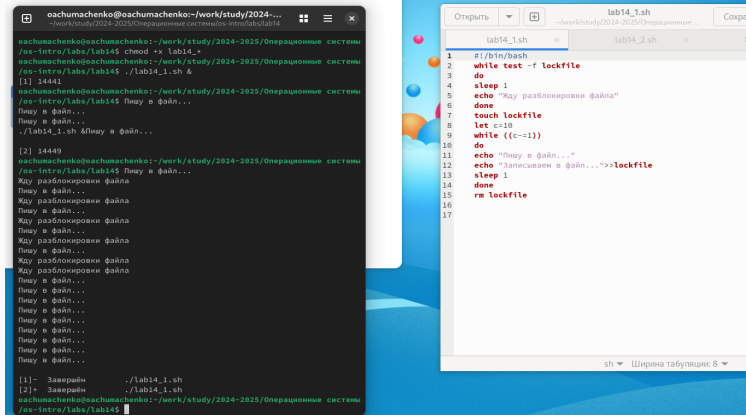
Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов

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 two windows from a Linux desktop environment. The left window is a terminal with the following content:

```
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ chmod +x lab14_*
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_1.sh &
[1] 14441
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ПИШУ в файл...
ПИШУ в файл...
./lab14_1.sh &ПИШУ в файл...

[2] 14449
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ 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
ПИШУ в файл...
ПИШУ в файл...
ПИШУ в файл...
ПИШУ в файл...
ПИШУ в файл...
ПИШУ в файл...
ПИШУ в файл...
ПИШУ в файл...

[1]- Завершил ./lab14_1.sh
[2]+ Завершил ./lab14_1.sh
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$
```

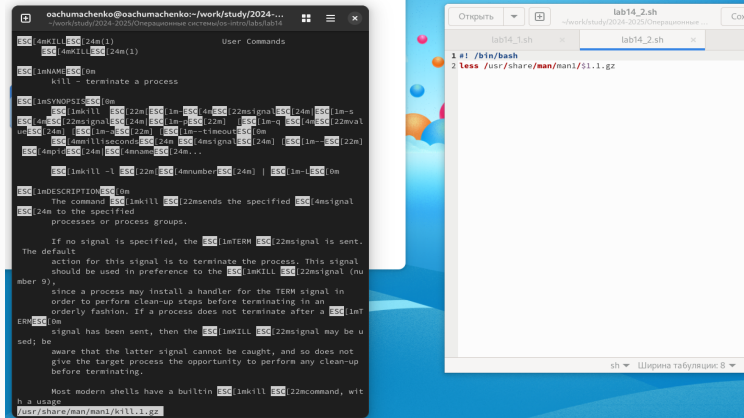
The right window is a script editor titled 'lab14_1.sh' containing the following code:

```
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--))
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 terminal windows. The left window displays the manual page for the `kill` command, which explains that it is used to terminate a process. It details the use of signals, with `SIGTERM` (signal 15) being the default for graceful termination and `SIGKILL` (signal 9) being used for forced termination. The right window shows a terminal session where the `kill` command is used to terminate a process named `man1`. The command `kill -l` is used to list the signals, and `kill -9 man1` is used to send the `SIGKILL` signal to the process.

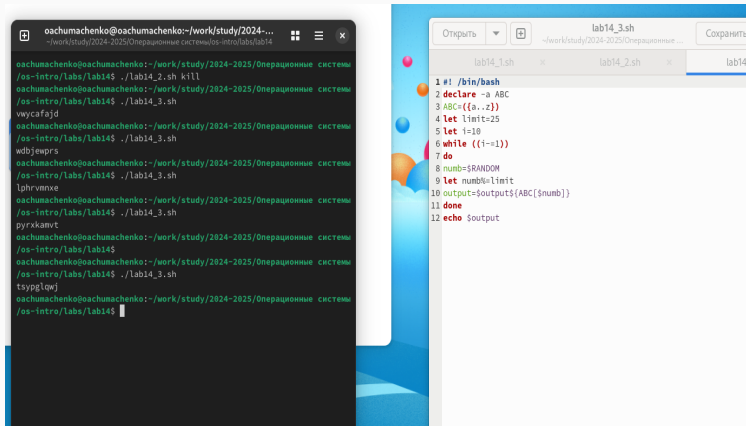
```
oachumachenko@oachumachenko:~/work/study/2024-...  
~/.work/study/2024-2025/Операционные системы/os-intro/labs/lab14  
ESC[4mKILLESC[24m(1) User Commands  
ESC[4mKILLESC[24m(1)  
ESC[1mNAMEESC[0m  
kill - terminate a process  
ESC[1mSYNOPSISESC[0m  
ESC[1mkill ESC[22mESC[1m-ESC[4mESC[22msignalESC[24mESC[1m-s  
ESC[4mESC[22msignalESC[24mESC[1m-pESC[22m] ESC[1m-q ESC[4mESC[22mval  
ueESC[24m] ESC[1m-pESC[22m] ESC[1m--timeoutESC[0m  
ESC[4mkillsecondsESC[24mESC[4msignalESC[24m] ESC[1m--ESC[22m]  
ESC[4mpidESC[24m]ESC[4mnameESC[24m...  
ESC[1mkill -l ESC[22mESC[4mnumberESC[24m] | ESC[1m-ESC[0m  
ESC[1mDESCRIPTIONESC[0m  
The command ESC[1mkill ESC[22msends the specified ESC[4msignal  
ESC[24m to the specified  
processes or process groups.  
If no signal is specified, the ESC[1mTERM ESC[22msignal is sent.  
The default  
action for this signal is to terminate the process. This signal  
should be used in preference to the ESC[1mKILL ESC[22msignal (nu  
mber 9),  
since a process may install a handler for the TERM signal in  
order to perform clean-up steps before terminating in an  
orderly fashion. If a process does not terminate after a ESC[1mT  
ERMESC[0m  
signal has been sent, then the ESC[1mKILL ESC[22msignal may be u  
sed; be  
aware that the latter signal cannot be caught, and so does not  
give the target process the opportunity to perform any clean-up  
before terminating.  
Most modern shells have a builtin ESC[1mkill ESC[22mcommand, wit  
h a usage  
/usr/share/man/man1/kill.1.gz
```

```
lab14_2.sh  
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 displays two terminal windows side-by-side. The left window shows a series of commands being executed in a shell script, with the user's name 'oachumachenko' and the current directory path visible at the top. The right window shows a shell script named 'lab14_3.sh' with various commands including 'declare', 'let', 'while', 'do', 'done', and 'echo'.

```
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_2.sh kill
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
vuycafajd
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
wdbjewprs
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
lphrvnxkx
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
pyrxkamvt
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/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.