

Операционные системы

Программирование в командном процессоре ОС UNIX.

Вишняков Родион Сергеевич

19 июня 2025

Российский университет дружбы народов, Москва, Россия

Цели и задачи работы

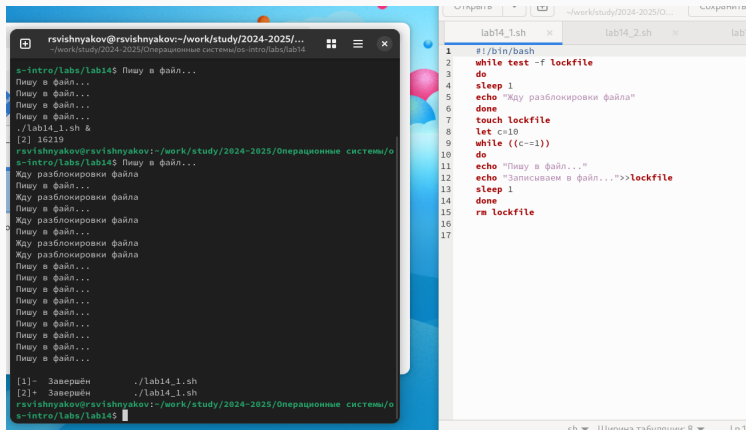
Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов

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 code editor on the right. The terminal window displays the execution of a script named `lab14_1.sh`. The script's output consists of alternating prompts to write to a file and wait for file unlocking, repeated 10 times. The code editor on the right shows the source code of `lab14_1.sh`, which implements a file locking mechanism using `lockfile` and `sleep`.

```
rsvishnyakov@rsvishnyakov:~/work/study/2024-2025/...  
~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14  
  
s-intro/labs/lab14$ ПISHУ в файл...  
ПISHУ в файл...  
ПISHУ в файл...  
ПISHУ в файл...  
./lab14_1.sh &  
[2] 16219  
  
rsvishnyakov@rsvishnyakov:~/work/study/2024-2025/Операционные системы/o  
s-intro/labs/lab14$ ПISHУ в файл...  
Жду разблокировки файла  
ПISHУ в файл...  
Жду разблокировки файла  
ПISHУ в файл...  
Жду разблокировки файла  
ПISHУ в файл...  
Жду разблокировки файла  
ПISHУ в файл...  
Жду разблокировки файла  
ПISHУ в файл...  
ПISHУ в файл...  
ПISHУ в файл...  
ПISHУ в файл...  
ПISHУ в файл...  
ПISHУ в файл...  
ПISHУ в файл...  
ПISHУ в файл...  
ПISHУ в файл...  
ПISHУ в файл...  
  
[1]- Завершён ./lab14_1.sh  
[2]+ Завершён ./lab14_1.sh  
rsvishnyakov@rsvishnyakov:~/work/study/2024-2025/Операционные системы/o  
s-intro/labs/lab14$
```

```
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 "ПISHУ 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

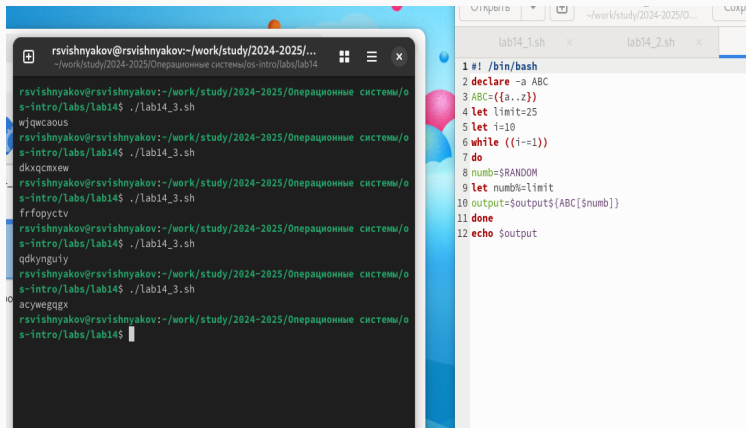
The image shows a terminal window on the left and a file editor on the right. The terminal window, titled 'rsvishnyakov@rsvishnyakov:~/work/study/2024-2025/...', shows the user typing 'df -h' and receiving the output of the command. The file editor, titled 'lab14_2.sh', shows a script with two lines: '1 #! /bin/bash' and '2 less /usr/share/man/man1/df.1.gz'.

```
rsvishnyakov@rsvishnyakov:~/work/study/2024-2025/...  
ESC[4mDFESC[24m(1) User Commands  
ESC[4mDFESC[24m(1)  
ESC[1mNAMEESC[0m  
df - report file system space usage  
14. ESC[1mSYNOPSISESC[0m  
ESC[1mdf ESC[22mESC[4mOPTIONESC[24m]... ESC[4mFILEESC[24m]...  
ESC[1mDESCRIPTIONESC[0m  
This manual page documents the GNU version of ESC[1mdfESC[22m.  
ESC[1mdf ESC[22mdisplays  
the amount of space available on the file system containing  
each file name argument. If no file name is given, the space  
available on all currently mounted file systems is shown.  
Space is shown in 1K blocks by default, unless the environment  
variable POSIXLY_CORRECT is set, in which case 512-byte blocks  
are used.  
If an argument is the absolute file name of a device node con-  
taining a mounted file system, ESC[1mdf ESC[22mshows the space a-  
vailable on  
that file system rather than on the file system containing the  
device node. This version of ESC[1mdf ESC[22mcannot show the s-  
pace avail-  
able on unmounted file systems, because on most kinds of sys-  
tems doing so requires non-portable intimate knowledge of file  
system structures.  
ESC[1mOPTIONSESC[0m  
/usr/share/man/man1/df.1.gz
```

```
lab14_2.sh  
~ /work/study/2024-2025/O...  
lab14_1.sh x lab14_2.sh x lab14_3  
1 #! /bin/bash  
2 less /usr/share/man/man1/df.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 script file on the right. The terminal window displays the execution of a script named `lab14_3.sh` in a directory `~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14`. The script's output consists of several lines of random lowercase letters. The script file on the right, `lab14_2.sh`, contains a Bash script that declares a variable `ABC` with the value `{a..z}`, sets a `limit` of 25, and uses a `while` loop to generate 10 random strings, each 25 characters long, using the `ABC` variable.

```
rsvishnyakov@rsvishnyakov:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
wjqwcacous
rsvishnyakov@rsvishnyakov:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
dkxqcmxew
rsvishnyakov@rsvishnyakov:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
frfopyctv
rsvishnyakov@rsvishnyakov:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
qdkynguiy
rsvishnyakov@rsvishnyakov:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
acywegqgx
rsvishnyakov@rsvishnyakov:~/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.