

Операционные системы

Программирование в командном процессоре ОС UNIX.

Гейдарова Назиля Рамиль кызы

22 апреля 2025

Российский университет дружбы народов, Москва, Россия

Цели и задачи работы

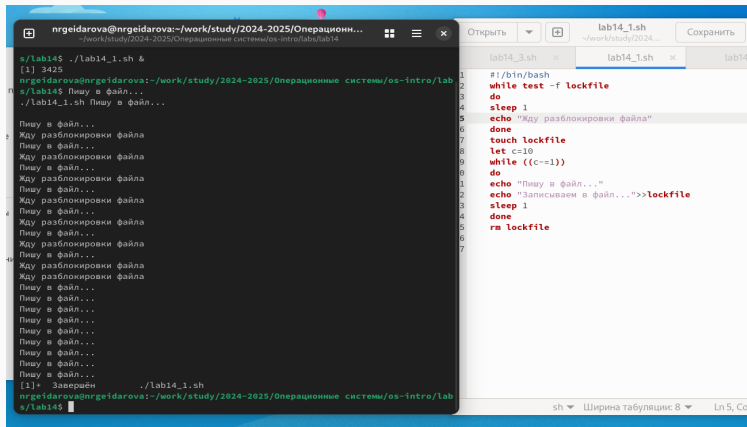
Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов

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 code editor on the right. The terminal window displays the execution of a script named `lab14_1.sh`. The script runs in the background, and the terminal output shows a series of messages: "Пишу в файл...", "Жду разблокировки файла", and "Пишу в файл...". The terminal window title is `nrgaidarova@nrgaidarova:~/work/study/2024-2025/Операционн...`. The code editor on the right shows the source code of `lab14_1.sh`. The script uses a `while` loop to wait for a file lock, indicated by the presence of a `lockfile`. The script uses `sleep 1` to wait for one second. The terminal window shows the script has completed its execution, indicated by the message "[1]+ Завершён ./lab14_1.sh".

```
s/lab14$ ./lab14_1.sh &
[1] 3425
nrgaidarova@nrgaidarova:~/work/study/2024-2025/Операционные системы/os-intro/lab$
s/lab14$ Пишу в файл...
./lab14_1.sh 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
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...
Пишу в файл...
[1]+  Завершён      ./lab14_1.sh
nrgaidarova@nrgaidarova:~/work/study/2024-2025/Операционные системы/os-intro/lab$
s/lab14$
```

```
#!/bin/bash
while test -f lockfile
do
sleep 1
echo "Жду разблокировки файла"
done
touch lockfile
let c=10
while ((c-=1))
do
echo "Пишу в файл..."
echo "Записываем в файл...">>lockfile
sleep 1
done
rm lockfile
```

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

```
nrgeidarova@nrgeidarova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14
ESC[4mPWDESC[24m(1)
ESC[4mPWDESC[24m(1)

ESC[1mNAMEESC[0m
pwd - print name of current/working directory

ESC[1mSYNOPSISESC[0m
ESC[1mpwd ESC[22m[ESC[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 symlinks

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-PESC[22mis assumed.

Your shell may have its own version of pwd, which usually supersedes the
version described here. Please refer to your shell's documentation for
details about the options it supports.

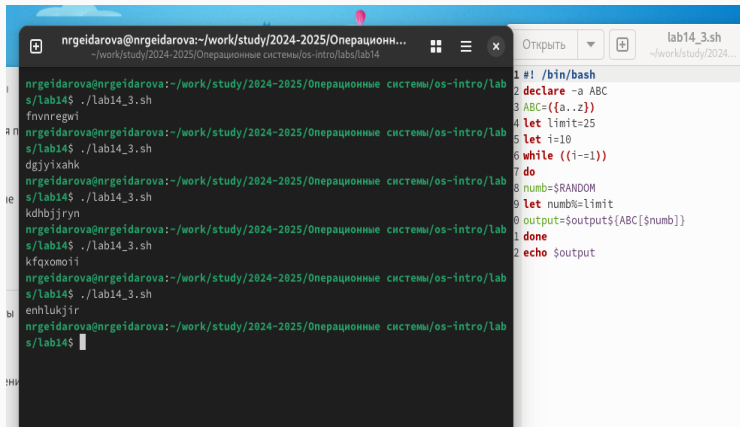
ESC[1mAUTHORESC[0m
Written by Jim Meyering.

/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 screenshot shows a terminal window with a dark background. The title bar indicates the user is 'nrgeidarova' and the current directory is '~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14'. The terminal shows the execution of a script named 'lab14_3.sh'. The script's output consists of several lines of random lowercase letters. To the right of the terminal, a snippet of the script's source code is visible, showing a bash script that declares a variable 'ABC', sets a limit, and uses a while loop to generate random characters.

```
nrgeidarova@nrgeidarova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
fnvnregwi
nrgeidarova@nrgeidarova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
dgjyixahk
nrgeidarova@nrgeidarova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
kdhbjjryn
nrgeidarova@nrgeidarova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
kfqxomoi
nrgeidarova@nrgeidarova:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14$ ./lab14_3.sh
enhlukjir
nrgeidarova@nrgeidarova:~/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.