

Программирование в командном процессоре ОС UNIX.

Мартынов Николай Алексеевич НБИбд-02-21¹

26 мая, 2022, 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

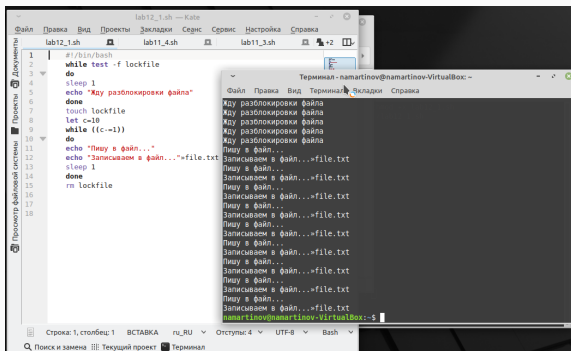
Задачи лабораторной работы

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 screenshot displays a Linux desktop environment. In the background, a text editor window titled 'lab12_1.sh — Kate' shows a shell script. The script is a loop that checks for a 'lockfile'. If it exists, it sleeps for 1 second and prints 'Жду разблокировки файла' (Waiting for file unblocking). If it doesn't exist, it creates the file, prints 'Пишу в файл...' (Writing to file...), and enters a loop where it repeatedly prints 'Записываем в файл...file.txt' (Writing to file...file.txt) and sleeps for 1 second. The script ends by removing the lockfile. In the foreground, a terminal window titled 'Терминал - namartinov@namartinov-VirtualBox: ~' shows the output of the script. It displays the waiting message followed by multiple instances of the writing message, indicating the script is running. The terminal prompt is 'namartinov@namartinov-VirtualBox:~\$'.

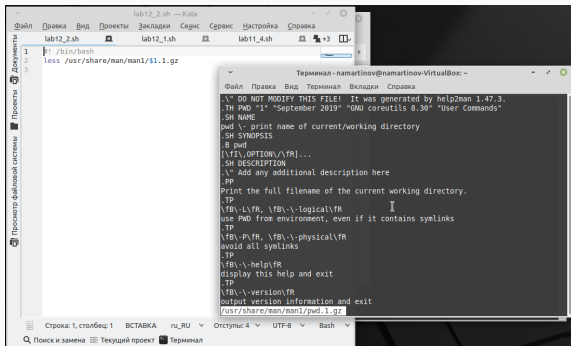
```
#!/bin/bash
while test -f lockfile
do
sleep 1
echo "Жду разблокировки файла"
done
touch lockfile
let c=10
while ((c-->1))
do
echo "Пишу в файл..."
echo "Записываем в файл...file.txt"
sleep 1
done
rm lockfile
```

Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Пишу в файл...
Записываем в файл...file.txt
Пишу в файл...
Записываем в файл...file.txt
Пишу в файл...
Записываем в файл...file.txt
Пишу в файл...
Записываем в файл...file.txt
Пишу в файл...
Записываем в файл...file.txt
Пишу в файл...
Записываем в файл...file.txt
Пишу в файл...
Записываем в файл...file.txt
Пишу в файл...
Записываем в файл...file.txt
Пишу в файл...
Записываем в файл...file.txt
namartinov@namartinov-VirtualBox:~\$

Figure 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 screenshot shows a terminal window titled 'lab12_2.sh — Kate'. The terminal content is as follows:

```
1 /bin/bash
2 less /usr/share/man/man1/$1.1.gz
3
```

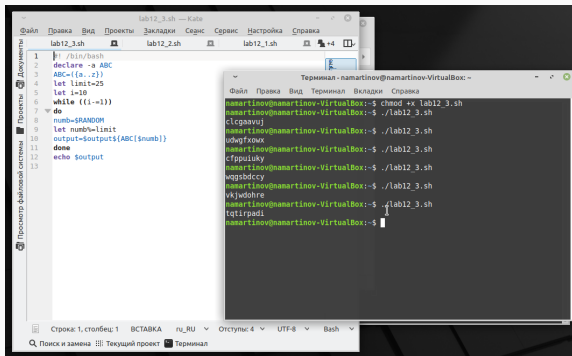
Below the terminal window, a separate window titled 'Терминал - namartinov@namartinov-VirtualBox: -' displays the output of the script. The output is a shell script template for 'pwd' (print working directory). The script includes comments in Russian and various options for displaying the full filename, logical/physical paths, and version information.

```
.\^ DO NOT MODIFY THIS FILE! It was generated by help2man 1.47.3.
.TH PWD "1" "September 2019" "GNU coreutils 8.30" "User Commands"
.SH NAME
pwd \- print name of current/working directory
.SH SYNOPSIS
.B pwd
[\[YFV\]OPTION\[\fR\]...]
.SH DESCRIPTION
.\^ Add any additional description here
.PP
Print the full filename of the current working directory.
.TP
\[\fB\]-L\[\fR], \[\fB\]-l\[\fR], \[\fB\]-L\[\fR]...
use PWD from environment, even if it contains symlinks
.TP
\[\fB\]-P\[\fR], \[\fB\]-p\[\fR], \[\fB\]-P\[\fR]...
avoid all symlinks
.TP
\[\fB\]-\[\fR]-help\[\fR]
display this help and exit
.TP
\[\fB\]-\[\fR]-version\[\fR]
output version information and exit
/usr/share/man/man1/pwd.1.gz
```

Figure 2: Задание 2

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

Выполнение работы



The screenshot displays a KVM virtual machine interface. On the left, a sidebar shows the file system structure with 'Documents' and 'Projects' sections. The main area is split into two panes. The top pane, titled 'lab12_3.sh', shows a shell script being edited. The script is as follows:

```
1 #! /bin/bash
2 declare -a ABC
3 ABC={a..z}
4 let limit=25
5 let i=10
6 while ((i-->0))
7 do
8   numb=$((RANDOM))
9   let numb%limit
10  output=$(cat /dev/urandom | tr -dc 'a-z' | fold -w $numb | xargs echo)
11  done
12 echo $output
```

The bottom pane is a terminal window titled 'Терминал - namartinov@namartinov-VirtualBox: ~'. It shows the execution of the script. The user runs 'chmod +x lab12_3.sh' and then repeatedly runs './lab12_3.sh', which outputs random strings of lowercase letters and numbers. The terminal output is as follows:

```
namartinov@namartinov-VirtualBox:~$ chmod +x lab12_3.sh
namartinov@namartinov-VirtualBox:~$ ./lab12_3.sh
clcgavuj
udwgfxowx
cfppuikuy
wqgsbdccy
vkjwdohre
tqirpadi
```

Figure 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.