

Программирование в командном процессоре ОС UNIX.

Роберт Кармазян ¹

21 апреля, 2024, 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

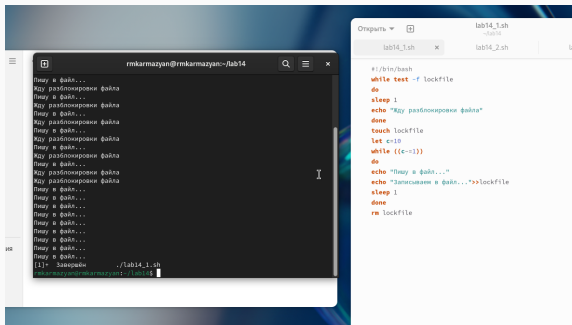
Задачи лабораторной работы

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



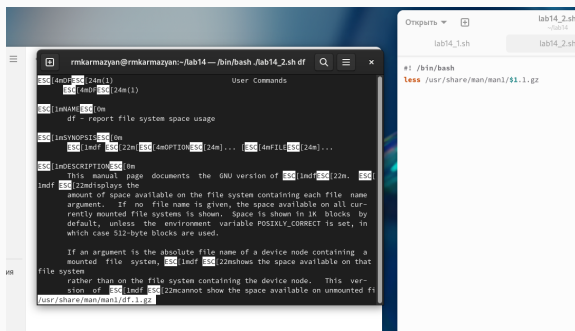
The image shows a terminal window on the left and a code editor on the right. The terminal window, titled 'rmkarmazyan@rmkarmazyan:~/lab14', displays the output of a script where the message 'Иду разблокировки файла' (Going to file unlocking) is printed repeatedly. The code editor, titled 'lab14_1.sh', shows the following script:

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



The image shows a terminal window and a file manager. The terminal window, titled 'rmkarmazyan@rmkarmazyan:~/lab14 — /bin/bash ./lab14_2.sh df', displays the output of the 'df' command. The output shows the disk space usage for the root filesystem (/) and the /usr/share/man directory. The file manager, titled 'lab14_2.sh', shows the contents of the script, which is a simple shell script that runs '/bin/bash' and then 'tess /usr/share/man/man1/df.1.gz'.

```
rmkarmazyan@rmkarmazyan:~/lab14 — /bin/bash ./lab14_2.sh df
User Commands
ESC[4mOfESC[24m(1)
ESC[4mOfESC[24m(1)

ESC[1mNAMEESC[0m
df - report file system space usage

ESC[1mSYNOPSISESC[0m
ESC[1mdfESC[22mESC[4mOPTIONESC[24m...ESC[4mFileESC[24m...

ESC[1mDESCRIPTIONESC[0m
This manual page documents the GNU version of ESC[1mdfESC[22m. ESC[1m
ESC[1mESC[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 cur-
rently 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.

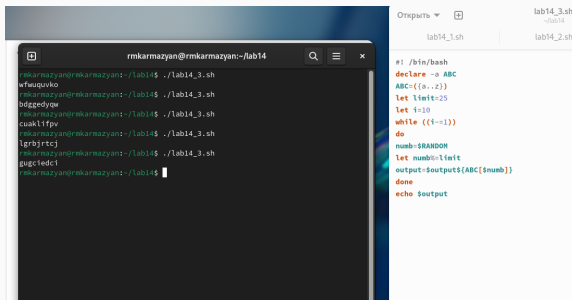
If an argument is the absolute file name of a device node containing a
mounted file system, ESC[1mdfESC[22mshows the space available on that
file system
rather than on the file system containing the device node. This ver-
sion of ESC[1mdfESC[22mcannot show the space available on unmounted fi
/usr/share/man/man1/df.1.gz
```

```
lab14_2.sh
#!/bin/bash
tess /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

Выполнение работы



The image shows a terminal window on the left and a code editor on the right. The terminal window has a title bar 'rmkarmazyan@rmkarmazyan:~/lab14' and shows the execution of a script 'lab14_3.sh' multiple times, producing random alphanumeric strings. The code editor on the right shows the content of 'lab14_3.sh'.

```
rmkarmazyan@rmkarmazyan:~/lab14$ ./lab14_3.sh
wfuuquvko
rmkarmazyan@rmkarmazyan:~/lab14$ ./lab14_3.sh
bdggedyqw
rmkarmazyan@rmkarmazyan:~/lab14$ ./lab14_3.sh
cuakl1fpv
rmkarmazyan@rmkarmazyan:~/lab14$ ./lab14_3.sh
lgrbjrttcj
rmkarmazyan@rmkarmazyan:~/lab14$ ./lab14_3.sh
gugciedci
rmkarmazyan@rmkarmazyan:~/lab14$
```

```
#!/bin/bash
declare -a ABC
ABC=([a..z])
let limit=25
let i=10
while ((i--))
do
numb=$RANDOM
let numb<limit
output=$output${ABC[$numb]}
done
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.