

Программирование в командном процессоре ОС UNIX.

Алина Тарасова¹

19 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

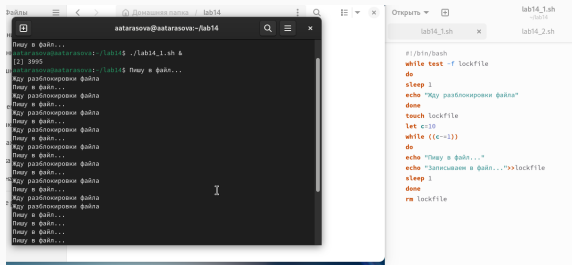
Задачи лабораторной работы

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 and a file editor. The terminal window, titled 'aatarasova@aatarasova:~lab14', shows a series of commands and outputs. The file editor, titled 'lab14_1.sh', shows a script for file locking.

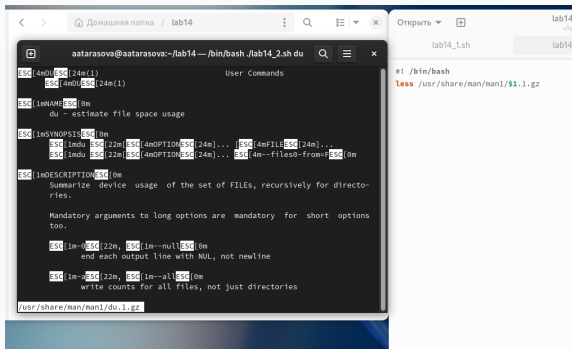
```
aatarasova@aatarasova:~lab14$ ./lab14_1.sh &
[2] 3995
aatarasova@aatarasova:~lab14$ Путь в файл...
Ключ разблокировки файла
Путь в файл...
Ключ разблокировки файла
Путь в файл...
Ключ разблокировки файла
Путь в файл...
Ключ разблокировки файла
Путь в файл...
Ключ разблокировки файла
Путь в файл...
Ключ разблокировки файла
Путь в файл...
Ключ разблокировки файла
Путь в файл...
Ключ разблокировки файла
Путь в файл...
Ключ разблокировки файла
Путь в файл...
Ключ разблокировки файла
Путь в файл...
```

```
#!/bin/bash
while test -f lockfile
do
sleep 1
echo "Ключ разблокировки файла"
done
touch lockfile
let c+=10
while ((c>0))
do
echo "Путь 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



The image shows a terminal window and a file manager side-by-side. The terminal window, titled 'aatarasova@aatarasova:~/lab14 --- /bin/bash ./lab14_2.sh du', displays the help text for the 'du' command. The file manager, titled 'lab14_1.sh', shows a file named 'lab14_1.sh' with the following content:

```
#!/bin/bash
less /usr/share/man/man1/du.1.gz
```

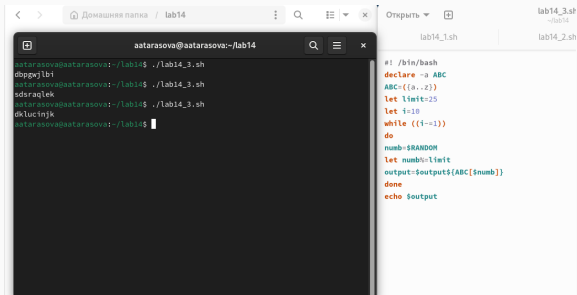
The terminal window shows the following text:

```
User Commands
ESC[4mduESC[24m(1)
ESC[4mduESC[24m(1)
ESC[1mNAMEESC[0m
du - estimate file space usage
ESC[1mSYNOPSISESC[0m
ESC[1mduESC[22mESC[4mOPTIONESC[24m)... ESC[4mFILEESC[24m)...
ESC[1mduESC[22mESC[4mOPTIONESC[24m)... ESC[4m--files0-from=ESC[0m
ESC[1mDESCRIPTIONESC[0m
Summarize device usage of the set of FILES, recursively for directories.
Mandatory arguments to long options are mandatory for short options too.
ESC[1m--0ESC[22m, ESC[1m--nullESC[0m
end each output line with NUL, not newline
ESC[1m--0ESC[22m, ESC[1m--allESC[0m
write counts for all files, not just directories
/usr/share/man/man1/du.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 and a file editor side-by-side. The terminal window, titled 'aatarasova@aatarasova: ~/lab14', shows the execution of a script named 'lab14_3.sh'. The output of the script is a series of random characters: 'dbpgwj1bi', 'sdsraqlek', 'dklucinj', and 'dklucinj'. The file editor, titled 'lab14_1.sh', shows the content of the script. The script is a bash script that declares a variable 'ABC' with a random string of 25 characters, sets a limit of 25, and then prints the first 'limit' characters of 'ABC' using 'printf'.

```
aatarasova@aatarasova:~/lab14$ ./lab14_3.sh
dbpgwj1bi
aatarasova@aatarasova:~/lab14$ ./lab14_3.sh
sdsraqlek
aatarasova@aatarasova:~/lab14$ ./lab14_3.sh
dklucinj
aatarasova@aatarasova:~/lab14$
```

```
#!/bin/bash
declare -a ABC
ABC=({a..z})
let limit=25
let i=10
while ((i-->0))
do
    numb=$((RANDOM%26))
    let numb+=limit
    output+=${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.