

Программирование в командном процессоре ОС UNIX.

Арсений Февралёв¹

26 апреля, 2023, 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

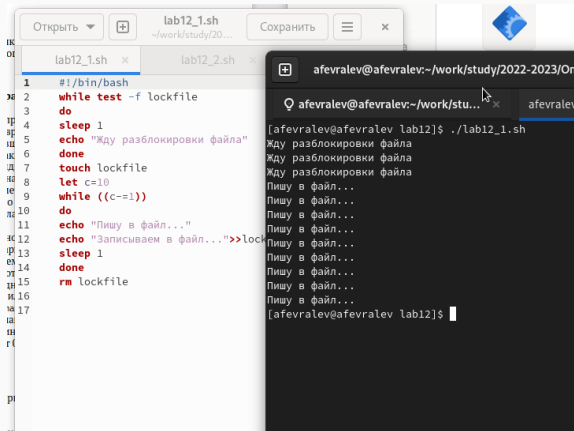
Задачи лабораторной работы

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 code editor window with a file named `lab12_1.sh` and a terminal window running the script.

Code Editor (lab12_1.sh):

```
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-->0))
10 do
11     echo "Пишу в файл..."
12     echo "Записываем в файл...">>lockfile
13     sleep 1
14 done
15 rm lockfile
```

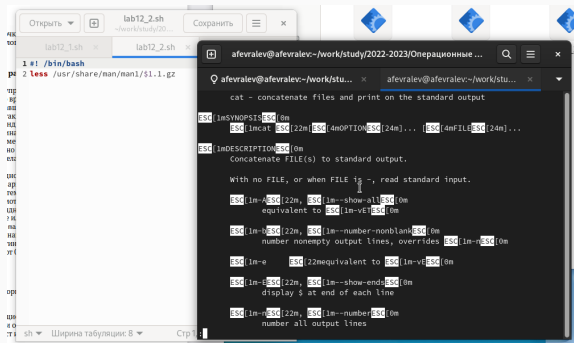
Terminal:

```
afevralev@afevralev:~/work/study/2022-2023/On...
[afevralev@afevralev lab12]$ ./lab12_1.sh
Жду разблокировки файла
Жду разблокировки файла
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...
Пишу в файл...
[afevralev@afevralev lab12]$
```

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

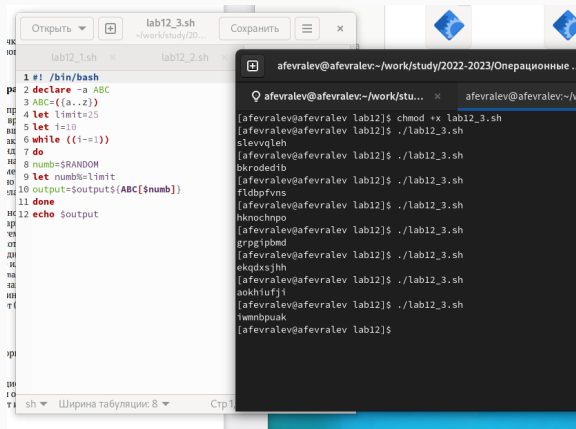


```
afevralev@afevralev-~/work/study/2022-2023/Операционные ...  
cat - concatenate files and print on the standard output  
  
ESC[1mSYNOPSISESC[0m  
ESC[1mcat ESC[22m[ESC[4mOPTIONESC[24m]... ESC[4mFILEESC[24m]...  
  
ESC[1mDESCRIPTIONESC[0m  
Concatenate FILE(s) to standard output.  
  
With no FILE, or when FILE is -, read standard input.  
  
ESC[1m-ASC[22m, ESC[1m--show-allESC[0m  
equivalent to ESC[1m-vESC[0m  
  
ESC[1m-bESC[22m, ESC[1m--number-nonblankESC[0m  
number nonempty output lines, overrides ESC[1m-nESC[0m  
  
ESC[1m-e ESC[22mequivalent to ESC[1m-vESC[0m  
  
ESC[1m-EESC[22m, ESC[1m--show-endsESC[0m  
display $ at end of each line  
  
ESC[1m-nESC[22m, ESC[1m--numberESC[0m  
number all output lines
```

Рис. 2: Задание 2

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

Выполнение работы



```
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
```

```
afevralev@afevralev:~/work/study/2022-2023/Операционные ...
afevralev@afevralev lab12]$ chmod +x lab12_3.sh
afevralev@afevralev lab12]$ ./lab12_3.sh
slevvgleh
afevralev@afevralev lab12]$ ./lab12_3.sh
bkrodedib
afevralev@afevralev lab12]$ ./lab12_3.sh
fldbpfvns
afevralev@afevralev lab12]$ ./lab12_3.sh
hknochnpo
afevralev@afevralev lab12]$ ./lab12_3.sh
grpgipbmd
afevralev@afevralev lab12]$ ./lab12_3.sh
ekqdxsjhh
afevralev@afevralev lab12]$ ./lab12_3.sh
aokhiuff1
afevralev@afevralev lab12]$ ./lab12_3.sh
1wmnbpuak
afevralev@afevralev lab12]$
```

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.