

Программирование в командном процессоре ОС UNIX.

Кенан Гашимов¹

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

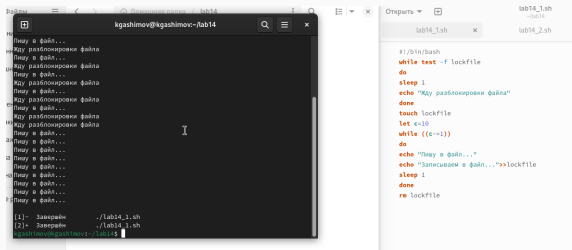
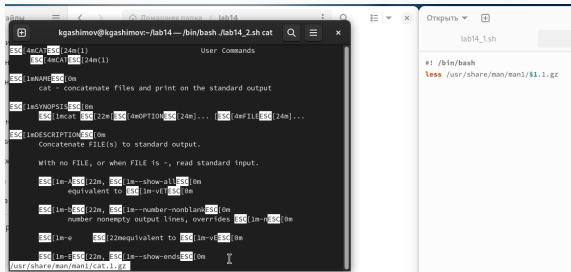


Рис. 1: Задание 1

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 browser window. The terminal window, titled 'kgashimov@kgashimov:~/lab14 — /bin/bash .lab14_2.sh cat', displays the help text for the 'cat' command. The file browser window, titled 'Открыть', shows the file path '/usr/share/man/man1/cat.1.gz'.

```
kgashimov@kgashimov:~/lab14 — /bin/bash .lab14_2.sh cat
ESC[4pCat ESC[24m(1)
ESC[4mCat ESC[24m(1)
ESC[1mNAME ESC[0m
cat - concatenate files and print on the standard output
ESC[1mSYNOPSIS ESC[0m
ESC[1mcat ESC[22m ESC[4mOPTION ESC[24m)... ESC[4mFILE ESC[24m)...
ESC[1mDESCRIPTION ESC[0m
Concatenate FILE(s) to standard output.
With no FILE, or when FILE is -, read standard input.
ESC[1m- ESC[22m, ESC[1m--show-all ESC[0m
equivalent to ESC[1m-v ESC[0m
ESC[1m-b ESC[22m, ESC[1m--number-nonblank ESC[0m
number nonempty output lines, overrides ESC[1m-n ESC[0m
ESC[1m-e ESC[22m equivalent to ESC[1m-v ESC[0m
ESC[1m-s ESC[22m, ESC[1m--show-ends ESC[0m
/usr/share/man/man1/cat.1.gz
```

Открыть

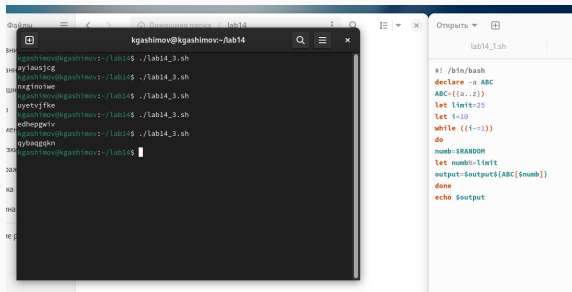
lab14_1.sh

#!/bin/bash
less /usr/share/man/man1/\$1.1.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 'kgashimov@kgashimov:~/.lab14', displays a series of commands being executed in a loop, each followed by a prompt. The commands are: `./lab14_3.sh`, `./lab14_3.sh`, `./lab14_3.sh`, `./lab14_3.sh`, `./lab14_3.sh`, `./lab14_3.sh`, `./lab14_3.sh`, and `./lab14_3.sh`. The file editor, titled 'lab14_1.sh', shows the content of the script being executed. The script is a shell script that declares a variable `ABC` with the value `{a..z}`, sets a `limit` of 25, and enters a `while` loop that runs 10 times. Inside the loop, a random number `numb` is generated, and the output of the `ABC` variable is printed `numb` times.

```
kgashimov@kgashimov:~/.lab14$ ./lab14_3.sh
kgashimov@kgashimov:~/.lab14$ ./lab14_3.sh
kgashimov@kgashimov:~/.lab14$ ./lab14_3.sh
kgashimov@kgashimov:~/.lab14$ ./lab14_3.sh
kgashimov@kgashimov:~/.lab14$ ./lab14_3.sh
kgashimov@kgashimov:~/.lab14$ ./lab14_3.sh
kgashimov@kgashimov:~/.lab14$ ./lab14_3.sh
kgashimov@kgashimov:~/.lab14$
```

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