

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Доленко Дарья Васильевна НБИбд-01-21¹

28 мая, 2022, Москва, Россия

¹Российский Университет Дружбы Народов

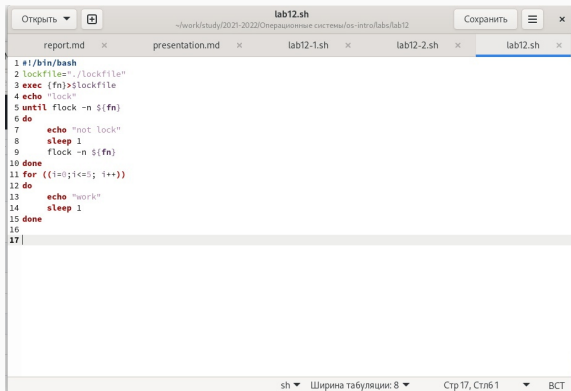
Цели и задачи работы

Цель лабораторной работы

Изучить основы программирования в оболочке ОС UNIX.
Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Процесс выполнения лабораторной работы

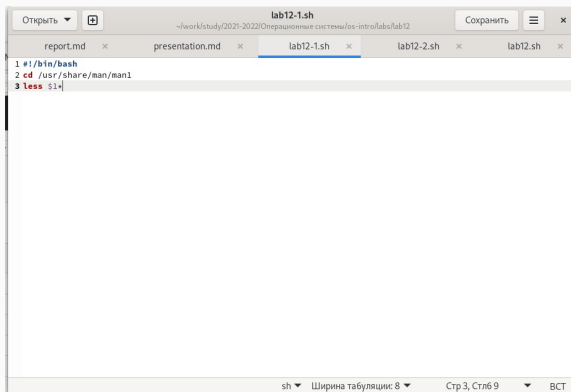
Написала командный файл, реализующий упрощённый механизм семафоров. (рис. 1)



```
#!/bin/bash
lockfile="/lockfile"
exec {fn}>$lockfile
echo "lock"
until flock -n ${fn}
do
    echo "not lock"
    sleep 1
    flock -n ${fn}
done
for ((i=0; i<=5; i++))
do
    echo "work"
    sleep 1
done
```

Figure 1: Текст программы

Реализовала команду man с помощью командного файла. (рис. 2 3)



The image shows a terminal window titled "lab12-1.sh" with a file path of "~/work/study/2021-2022/Операционные системы/os-intro/labs/lab12". The window has several tabs: "report.md", "presentation.md", "lab12-1.sh" (active), "lab12-2.sh", and "lab12.sh". The terminal content shows a shell script with three lines: 1. `#!/bin/bash`, 2. `cd /usr/share/man/man1`, and 3. `less $1`. The prompt is `$1`. The bottom status bar shows "sh", "Ширина табуляций: 8", "Стр 3, Стлб 9", and "ВСТ".

```
#!/bin/bash
cd /usr/share/man/man1
less $1
```

Figure 2: Текст программы

```
dvoldenko@fedora:~/work/study/2021-2022/Операционные с...
LESS(1)                                General Commands Manual                                LESS(1)

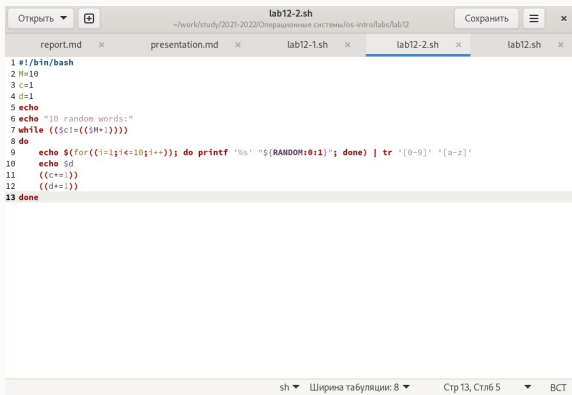
ESC[1mNAMEESC[0m
    less - opposite of more

ESC[1mSYNOPSISESC[0m
    ESC[1mless -?ESC[0m
    ESC[1mless --helpESC[0m
    ESC[1mless -VESC[0m
    ESC[1mless --versionESC[0m
    ESC[1mless [-[+]aABcCdeEfgGhIJKLmMnNqQrRsSuUVvWwX~]ESC[0m
    ESC[1m[-b ESC[4mESC[22mspaceESC[24mESC[1m] [-h ESC[4mESC[22mlinesESC[24mESC[1m] [-j ESC[4mESC[22mlineESC[24mESC[1m] [-k ESC[4mESC[22mkeyfileESC[24mESC[1m]ESC[0m
    ESC[1m[-o] ESC[4mESC[22mlogfileESC[24mESC[1m] [-p ESC[4mESC[22mpat
ternESC[24mESC[1m] [-P ESC[4mESC[22mpromptESC[24mESC[1m] [-t ESC[4mESC[22mtagESC[24mESC[1m]ESC[0m
    ESC[1m[-T ESC[4mESC[22mtagsfileESC[24mESC[1m] [-x ESC[4mESC[22mtabESC[24mESC[1m,...] [-y ESC[4mESC[22mlinesESC[24mESC[1m] [-z] ESC[4mESC[22mlinesESC[24mESC[1m]ESC[0m
    ESC[1m[-# ESC[4mESC[22mshiftESC[24mESC[1m] [+][+]ESC[4mESC[22mcmdESC[24mESC[1m] [--] ESC[4mESC[22mfilenameESC[24mESC[1m]...ESC[0m
    (See the OPTIONS section for alternate option syntax with long option
less.1.gz (file 1 of 3)
```

Figure 3: Результат

Используя встроенную переменную \$RANDOM, написала командный файл, генерирую-

щий случайную последовательность букв латинского алфавита. (рис. 4)



The screenshot shows a terminal window titled "lab12-2.sh" with a file explorer interface at the top. The file explorer shows several files: report.md, presentation.md, lab12-1.sh, lab12-2.sh (selected), and lab12.sh. The terminal content is a shell script that generates 10 random words. The script uses the \$RANDOM variable to generate random numbers and the tr command to convert them into random letters. The script is as follows:

```
1 #!/bin/bash
2 M=10
3 C=1
4 d=1
5 echo
6 echo "10 random words:"
7 while ((C!=($M+1)))
8 do
9     echo $(for((i=1;i<=10;i++)); do printf '%s' "${RANDOM:0:1}"; done) | tr '0-9' '[a-z]'
10    echo $d
11    ((C+=1))
12    ((d+=1))
13 done
```

At the bottom of the terminal window, there is a status bar showing "sh", "Ширина табуляций: 8", "Стр 13, Стлб 5", and "ВСТ".

Figure 4: Текст программы


```
[dvdolenko@fedora lab12]$ touch lab12.sh lab12-1.sh lab12-2.sh
[dvdolenko@fedora lab12]$ chmod +x *.sh
[dvdolenko@fedora lab12]$ ./lab12.sh
lock
work
work
work
work
work
work
[dvdolenko@fedora lab12]$ ./lab12-1.sh less
[dvdolenko@fedora lab12]$ ./lab12-2.sh

10 random words:
cccbcggcfc
1
cbbbceddbc
2
jhbcbfdgcb
3
```

Figure 5: Создание файлов, наделение их необходимыми правами и запуск.

Выводы по проделанной работе

В ходе данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX, научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.