

Лабораторная работа №12

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Галиева Аделина Руслановна

27 апреля 2023

Российский университет дружбы народов, Москва, Россия

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

1. Пишем командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени t_1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени $t_2 < t_1$, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запускаем командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой (`> /dev/tty#`, где `#` — номер терминала куда перенаправляется вывод), в котором также запущен этот файл, но не фоновом, а в привилегированном режиме.

```
argalievadk5n52 ~ $ cd ~/work/study/2022-2023/"Операционные системы"/os-intro/labs/lab12
argalievadk5n52 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ touch lab12.sh
argalievadk5n52 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ chmod +x lab12.sh
argalievadk5n52 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ ./lab12.sh
```

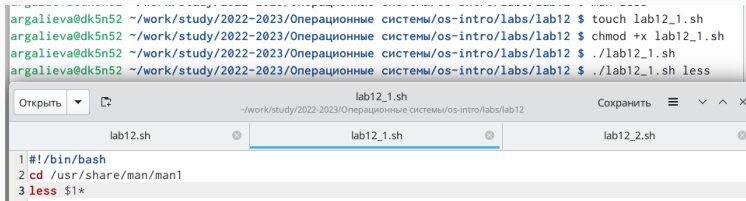


```
lab12.sh
1#!/bin/bash
2lockfile="./lockfile"
3exec {fn}>$lockfile
4exec "lock"
5until flock -n ${fn}
6do
7    echo "not lock"
8    sleep 1
9    flock -n ${fn}
10done
11for ((i=0;i<=5;i++))
12do
13    echo "work"
14    sleep 1
15done
```

Рис. 1: Задание 1

2. Реализуем команду `man` с помощью командного файла. Изучите содержимое каталога `/usr/share/man/man1`. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой `less` сразу же просмотрев содержимое справки. Командный файл должен получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге `man1`.

```
argalieva@dk5n52 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ touch lab12_1.sh
argalieva@dk5n52 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ chmod +x lab12_1.sh
argalieva@dk5n52 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ ./lab12_1.sh
argalieva@dk5n52 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ ./lab12_1.sh less
```



The screenshot shows a terminal window with a light gray background. The prompt is 'argalieva@dk5n52' and the current directory is '~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12'. The user has executed four commands: 'touch lab12_1.sh', 'chmod +x lab12_1.sh', './lab12_1.sh', and './lab12_1.sh less'. Below the terminal window, there is a window manager interface. The title bar of the window is 'lab12_1.sh' and the path is '~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12'. There are buttons for 'Открыть' (Open) and 'Сохранить' (Save). Below the title bar, there are three tabs: 'lab12.sh', 'lab12_1.sh' (which is active and highlighted with a blue underline), and 'lab12_2.sh'. The content of the active tab shows the first three lines of the script: '1 #!/bin/bash', '2 cd /usr/share/man/man1', and '3 less \$1*'. The text is color-coded: blue for the first line, red for the second line, and red for the third line.

Рис. 2: Задание 2

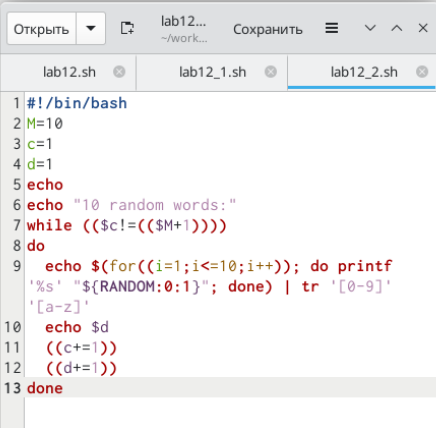
LESS(1)	General Commands Manual	LESS(1)
NAME		
	less - opposite of more	
SYNOPSIS		
	less -?	
	less --help	
	less -V	
	less --version	
	less [-[+]aABcCdeEfFgGiIJKLmMnNqQrRsSuUVvWX~]	
	[-b space] [-h lines] [-j line] [-k keyfile]	
	[-{o0} logfile] [-p pattern] [-P prompt] [-t tag]	
	[-T tagsfile] [-x tab,...] [-y lines] [-[z] lines]	
	[-# shift] [+ [+]cmd] [--] [filename]...	
	(See the OPTIONS section for alternate option syntax with long option names.)	
DESCRIPTION		

Рис. 3: Задание 3

3. Используем встроенную переменную \$RANDOM, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита.


```
argalieva@dk5n52 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12
$ touch lab12_2.sh
argalieva@dk5n52 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12
$ chmod +x lab12_2.sh
argalieva@dk5n52 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12
$ ./lab12_2.sh

10 random words:
gijccbccdc
1
cbgbbicdib
2
cbbbfccbgc
3
cjcccdcbdb
4
bcbdcdbcc
5
cbccbbbcc
6
bcccbefbcg
7
hbbcbgbffi
8
bbcbcedhcd
```



```
lab12.sh x lab12_1.sh x lab12_2.sh x
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 $(for((i=1;i<=10;i++)); do printf
10 '%s' "${RANDOM:0:1}"; done) | tr '0-9'
11 '[a-z]'
12 echo $d
13 ((c+=1))
14 ((d+=1))
15 done
```

Рис. 4: Задание 4

Я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.