

Программирование в командном процессоре ОС UNIX.

Диана Шмавонян¹

21 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

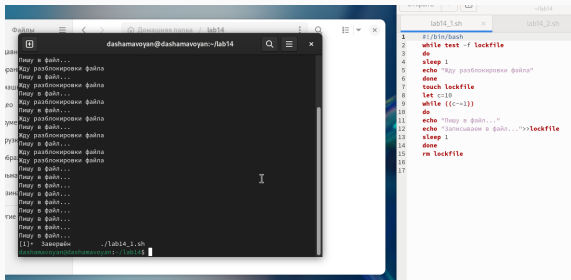
Задачи лабораторной работы

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 two windows from a Linux desktop environment. The left window is a terminal titled 'dashamavoyan@dashamavoyan: ~/lab14'. It displays the output of a script being executed, which consists of a loop printing 'Пишу в файл...' followed by 'Куда разблокирован файл' and then a series of numbers from 1 to 17. The right window is a code editor titled 'lab14_1.sh' and 'lab14_2.sh'. It shows the source code of the script, which is a shell script that creates a lockfile, enters a loop with a sleep command, updates the lockfile, and prints messages.

```
1 #!/bin/bash
2 while test -f lockfile
3 do
4     sleep 1
5     echo "Куда разблокирован файл"
6 done
7 touch lockfile
8 let c=10
9 while [[c--]]
10 do
11     echo "Пишу в файл..."
12     echo "Записываем в файл...">>lockfile
13     sleep 1
14 done
15 rm lockfile
16
17
```

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

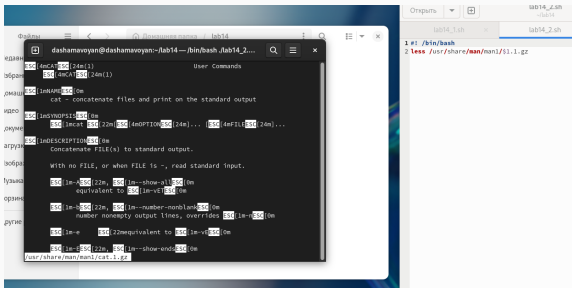
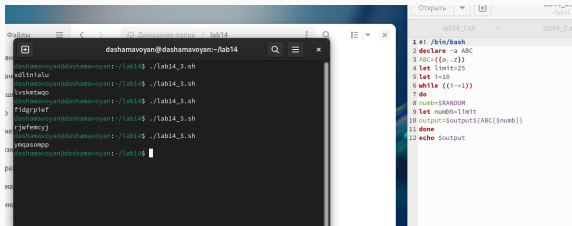


Рис. 2: Задание 2

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

Выполнение работы



The image shows a terminal window with a file explorer view. The terminal title is "dashaavoyan@dashaavoyan:~/lab14". The terminal content shows a list of files in the directory: "lab14_1.sh", "lab14_2.sh", "lab14_3.sh", "lab14_4.sh", "lab14_5.sh", "lab14_6.sh", "lab14_7.sh", "lab14_8.sh", "lab14_9.sh", "lab14_10.sh", "lab14_11.sh", "lab14_12.sh", "lab14_13.sh", "lab14_14.sh", "lab14_15.sh", "lab14_16.sh", "lab14_17.sh", "lab14_18.sh", "lab14_19.sh", "lab14_20.sh", "lab14_21.sh", "lab14_22.sh", "lab14_23.sh", "lab14_24.sh", "lab14_25.sh", "lab14_26.sh", "lab14_27.sh", "lab14_28.sh", "lab14_29.sh", "lab14_30.sh", "lab14_31.sh", "lab14_32.sh", "lab14_33.sh", "lab14_34.sh", "lab14_35.sh", "lab14_36.sh", "lab14_37.sh", "lab14_38.sh", "lab14_39.sh", "lab14_40.sh", "lab14_41.sh", "lab14_42.sh", "lab14_43.sh", "lab14_44.sh", "lab14_45.sh", "lab14_46.sh", "lab14_47.sh", "lab14_48.sh", "lab14_49.sh", "lab14_50.sh", "lab14_51.sh", "lab14_52.sh", "lab14_53.sh", "lab14_54.sh", "lab14_55.sh", "lab14_56.sh", "lab14_57.sh", "lab14_58.sh", "lab14_59.sh", "lab14_60.sh", "lab14_61.sh", "lab14_62.sh", "lab14_63.sh", "lab14_64.sh", "lab14_65.sh", "lab14_66.sh", "lab14_67.sh", "lab14_68.sh", "lab14_69.sh", "lab14_70.sh", "lab14_71.sh", "lab14_72.sh", "lab14_73.sh", "lab14_74.sh", "lab14_75.sh", "lab14_76.sh", "lab14_77.sh", "lab14_78.sh", "lab14_79.sh", "lab14_80.sh", "lab14_81.sh", "lab14_82.sh", "lab14_83.sh", "lab14_84.sh", "lab14_85.sh", "lab14_86.sh", "lab14_87.sh", "lab14_88.sh", "lab14_89.sh", "lab14_90.sh", "lab14_91.sh", "lab14_92.sh", "lab14_93.sh", "lab14_94.sh", "lab14_95.sh", "lab14_96.sh", "lab14_97.sh", "lab14_98.sh", "lab14_99.sh", "lab14_100.sh". The terminal also shows the command "ls" and the output of the command. The terminal also shows the command "cat lab14_2.sh" and the output of the command. The terminal also shows the command "echo" and the output of the command.

```
1 #!/bin/bash
2 declare -a ABC
3 ABC=(a..z)
4 let limit=25
5 let i=10
6 while ((i--))
7 do
8   numb=$((RANDOM))
9   let numb=$((numb % limit))
10  output=$(printf "%s" "${ABC:$numb}")
11 done
12 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.