Отчет по лабораторной работе №12

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Максим Сергеевич Белов

Содержание

# 1 Цель работы

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

# 2 Задание

1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени t1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени t2<>t1, также выдавая информацию о том, чторесурс используется соответствующим командным файлом (процессом). Запустить командный файл в одном виртуальном терминале в фоновом режиме, перенаправивего вывод в другой (> /dev/tty#, где # — номер терминала куда перенаправляется вывод), в котором также запущен этот файл, но не фоновом, а в привилегированномрежиме. Доработать программу так, чтобы имелась возможность взаимодействия трёхи более процессов.
2. Реализовать 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.
3. Используя встроенную переменную $RANDOM, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что $RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

# 3 Теоретическое введение

В табл. 1 приведено краткое описание команд для работы с командными файлами.

Таблица 1: Описание команд для работы с командными файлами

| Команда | Описание |
| --- | --- |
| getopts | Осуществляет синтаксический анализ командной строки, выделяя флаги, и используетсядля объявления переменных. |

# 4 Выполнение лабораторной работы

1. Скрипт 1 (Рис. 1)

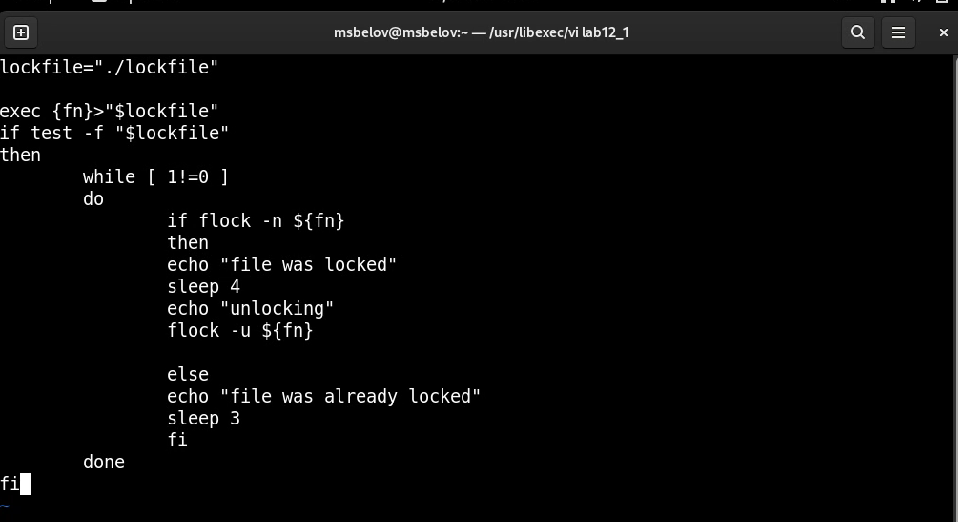


Рис. 1: Скрипт 1

Сделаем файл lab12\_1 исполняемым и проверим его работу (Рис. 2)

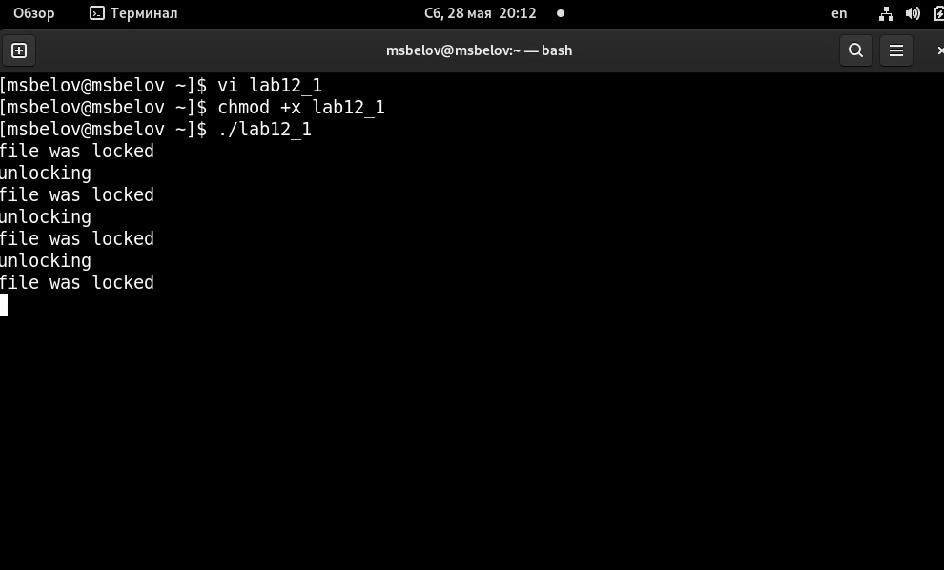


Рис. 2: Проверка скрипта 1

1. Скрипт 2 (Рис. 3)

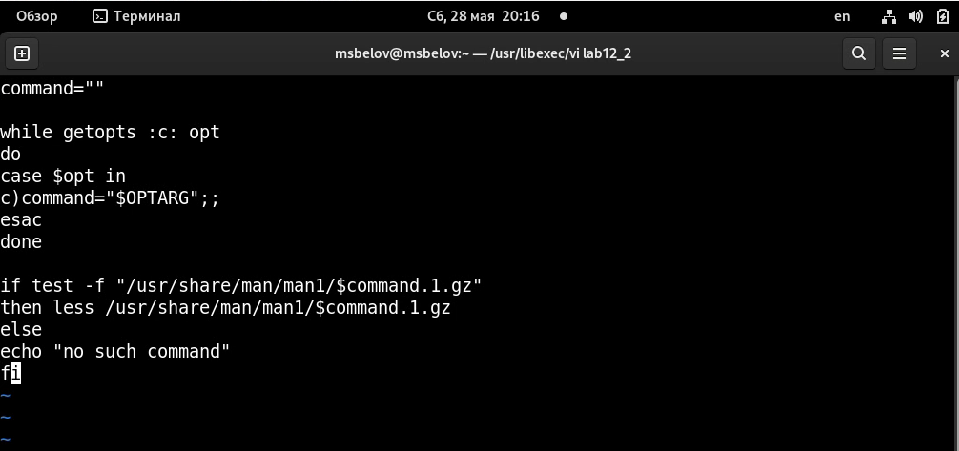


Рис. 3: Скрипт 2

Сделаем файл lab12\_2 исполняемым и проверим его работу (Рис. 4)

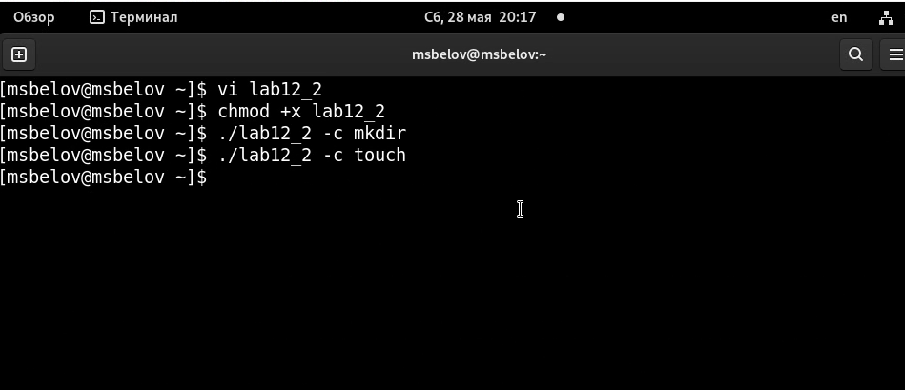


Рис. 4: Проверка скрипта 2

1. Скрипт 3 (Рис. 5)

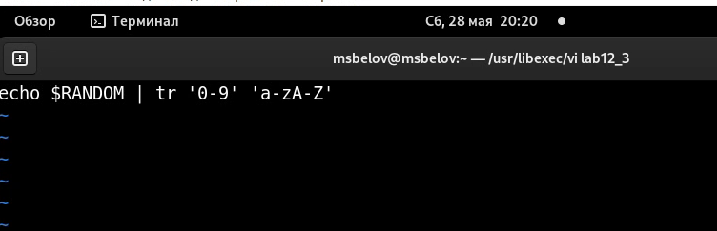


Рис. 5: Скрипт 3

Сделаем файл lab12\_3 исполняемым и проверим его работу

(Рис. 6)

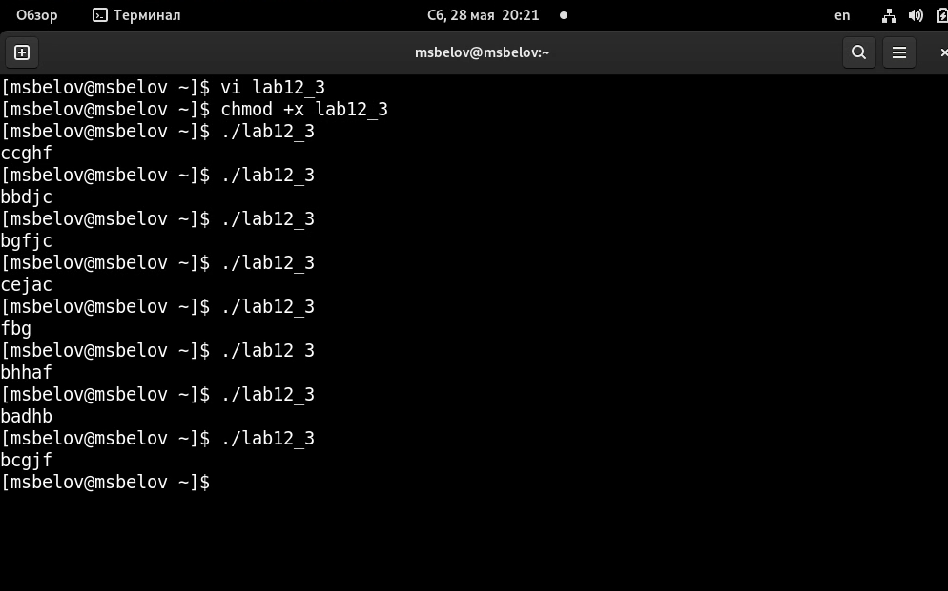


Рис. 6: Проверка скрипта 3

# 5 Выводы

В ходе работы я изучил основы программирования в оболочке ОС UNIX. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# 6 Контрольные вопросы

1. Между скобками должны быть пробелы. В противном случае скобки и рядом стоящие символы будут восприниматься как одно целое.
2. С помощью +=
3. seq - выдает последовательность чисел.
4. 5
5. Zsh очень сильно упрощает работу. Но существуют различия. Например, в zsh после for обязательно вставлять пробел, нумерация массивов в zsh начинается с 1 (что не особо удобно на самом деле).
6. Верен.
7. Bash позволяет очень легко работать с файловой системой без лишних конструкций (в отличи от обычного языка программирования). Но относительно обычных языков программирования bash очень сжат. Тот же Си имеет гораздо более широкие возможности для разработчика.