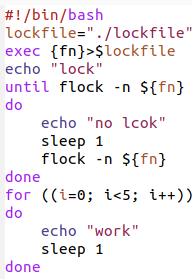
Лабораторная работа №12

Покрас Илья Михайлович

# Цель работы

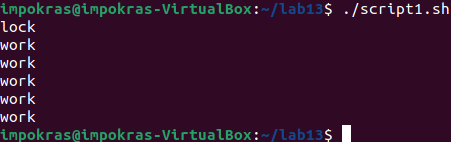
Изучить основы программирования в оболочке ОС UNIX, научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# Последовательность работы

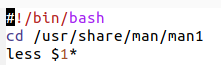


Script 1

Я написал 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, что ресурс используется соответствующим командным файлом. Запустить командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой, в котором также запущен этот файл, но не фоновом, а в привилегированном режиме.

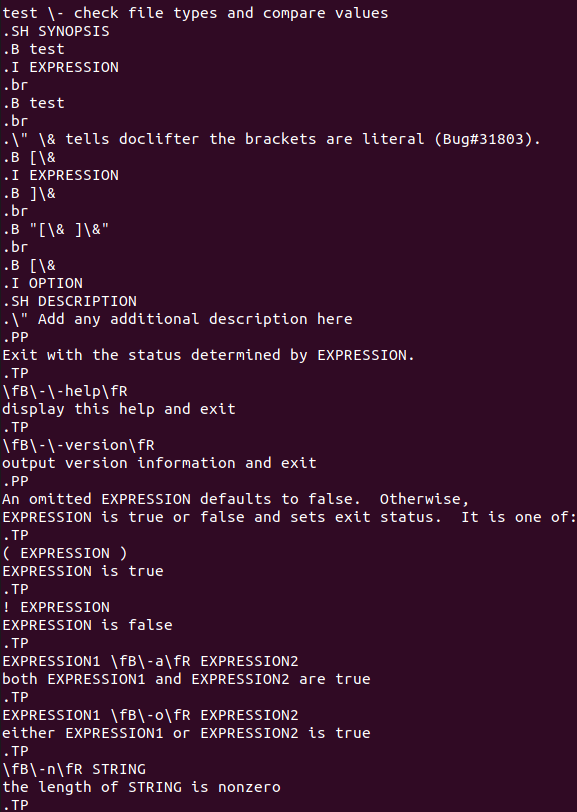


Script 1 result

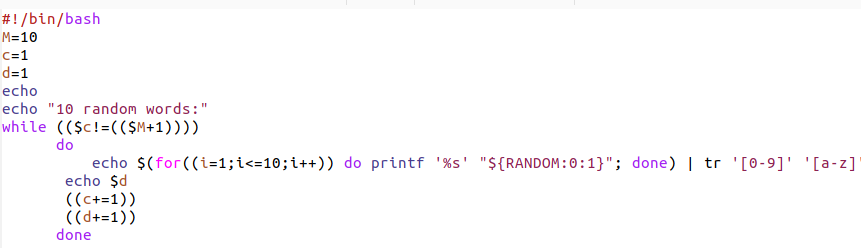


Script 2

Я реализовал команду man с помощью командного файла. Изучил 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

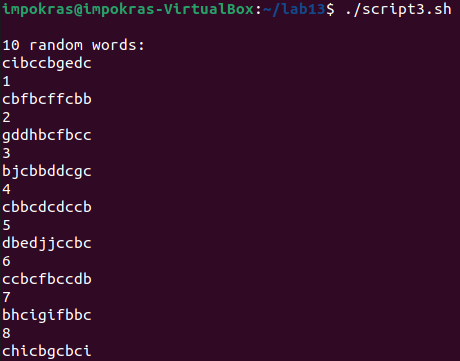


Script 2 result



Script 3

Используя встроенную переменную $RANDOM, я написал командный файл, генерирующий случайную последовательность букв латинского алфавита.



Script 3 result

# Вывод

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

# Контрольные вопросы:

1.Нужны пробелы после и перед открывающей и закрывающей скобками соответственно. Также желательно заключить $1 в кавычки (“$1”) во избежание ошибки, если str1$str2” 3.Seq выводит последовательность целых чисел с шагом, заданным пользователем. Другая утилита с той же функцией – jot. 4.Результатом будет 3. 5.В zsh можно настраивать горячие клавиши. Автодополнение более сложное и гибкое. Используется большое количество различных опций, а также максимально краткий синтаксис. В итоге, zsh удобен для повседневной, рутинной работы, а для написания скриптов всё же лучше использовать bash. 6.Синтаксис верен. 7.Bash имеет достаточно много сходств другими языками. Недостатком является достаточно нагруженный синтаксис (легко допустить ошибку, потеряв, допустим, fi). В целом он достаточно понятен, однако он выглядит не совсем обычно в некоторых моментах, также необходимо читать много справок, чтобы на нём писать.