Лабораторная работа №14

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Буллер Татьяна Александровна

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 Выполнение лабораторной работы

## 2.1 Семафоры

Необходимо написать командный файл, реализующий упрощённый механизм семафоров. В случае, если файд уже используется некоторым другим процессом, командный файл ожидает его освобождения 5 секунд и выдает соответствующее сообщение. Затем, если файл не освободился, цикл повторяется. Если же файл после ожидания стал свободен, то выводится сообщение о записи в файл и в сам файл записывается некоторая фраза.

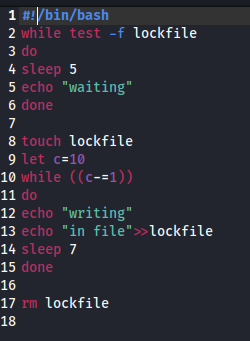


Рис. 1: Скрипт семафора

Проверим работу файла: откроем два окна терминала и в одном из них запустим файл в привелигированном режиме, а во втором - в фоновом, переведя вывод в первое окно. Для перевода вывода в окно графического терминала используем команду > /dev/pts/number, где number - номер графического интерфейса.

Перевод вывода в другое окно

Рис. 2: Перевод вывода в другое окно

Видим, что запущенный в первом окне файл первое время производит запись без проблем. Потом, когда подключается второй процесс, один из файлов начинает выводить сообщения об ожидании, когда файл записи оказывается занят.

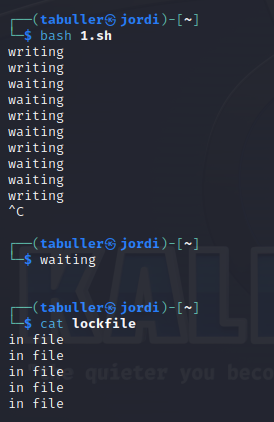


Рис. 3: Результат выполнения скрипта

## 2.2 Man с помощью командного файла

содержимое каталога /usr/share/man/man1 - архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. На некоторых дистрибутивах Linux каждый архив можно открыть командой less, сразу же просмотрев содержимое справки, однако в случае Kali эта опция less по умолчанию отключена. Для того, чтобы открыть архив .gz на Kali использую zcat и перевожу вывод в less. В качестве аргумента передаем название программы, которое вставится в код скрипта.

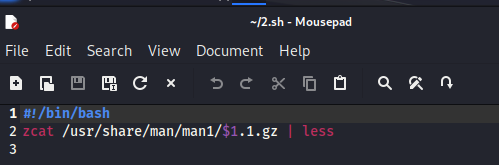


Рис. 4: Скрипт man

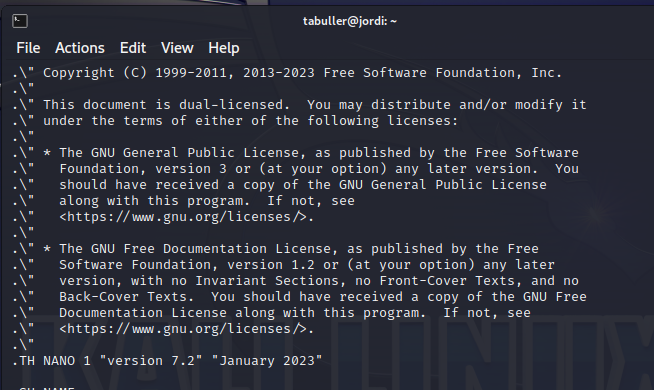


Рис. 5: Результат работы скрипта

## 2.3 Случайная комбинация

Зададим переменную alphabet: массив, который заполним латинскими маленькими буквами (a..z). Далее зададим переменную-ограничитель и проведем цикл по ней: на каждой итерации в переменную numb запишем случайное число, которое ограничим установленным до этого лимитом. Заполним полученными элементами переменную-массив вывода и вызовем ее в конце программы:

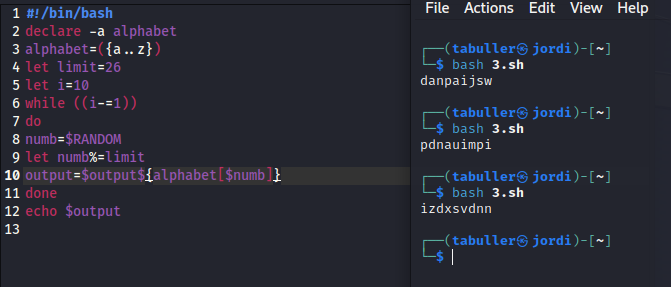


Рис. 6: Скрипт и результат выполнения

# 3 Выводы

Изучены основы программирования в оболочке ОС UNIX/Linux. Написаны более сложные командные файлы с использованием логических управляющих конструкций и циклов.