Лабораторная работа №14

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Кижваткина Анна Юрьевна

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

Создаем директорию lab14 и перемещаемся в неё. (рис. 1)

Создание директории

Рис. 1: Создание директории

Создаем файл 1.sh. (рис. 2)

Создание файла

Рис. 2: Создание файла

Пишем 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 (процессом). Запустить 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. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов. . (рис. 3)

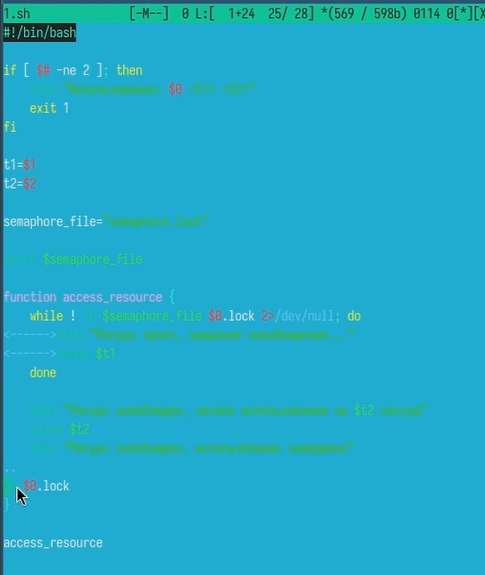


Рис. 3: Программа

Устанавливаем право на выполнение. (рис. 4)

Предоставление доступа

Рис. 4: Предоставление доступа

Проверяем выполнение программы. (рис. 5)

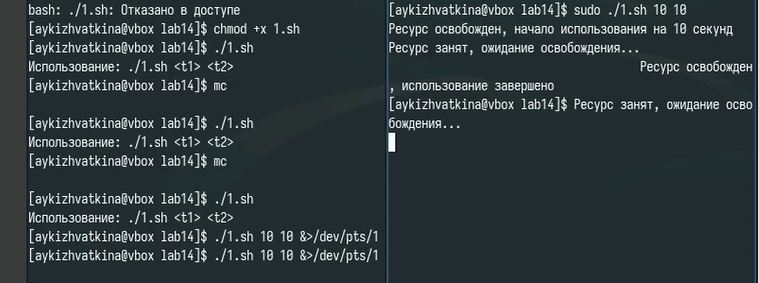


Рис. 5: Запуск программы

Создаем файл 2.sh и устанавливаем право на выполнение. (рис. 6)

Создание файла

Рис. 6: Создание файла

Реализовываем 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. (рис. 7 и рис. 8)

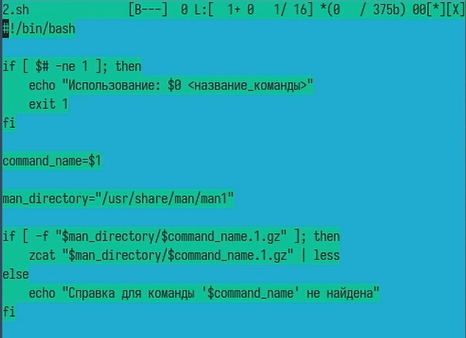


Рис. 7: Программа

Проверяем выполнение программы. (рис. 8)

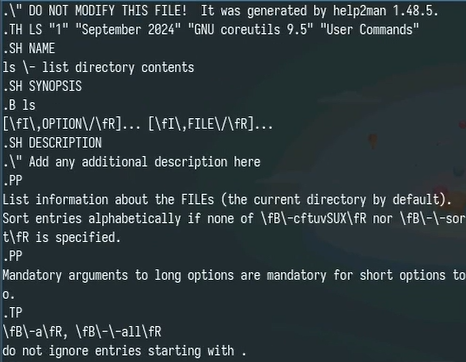


Рис. 8: Запуск программы

Создаем файл 3.sh и устанавливаем право на выполнение. (рис. 9)

Создание файла

Рис. 9: Создание файла

Используя встроенную переменную $RANDOM, пишем командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что $RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767. (рис. 10)

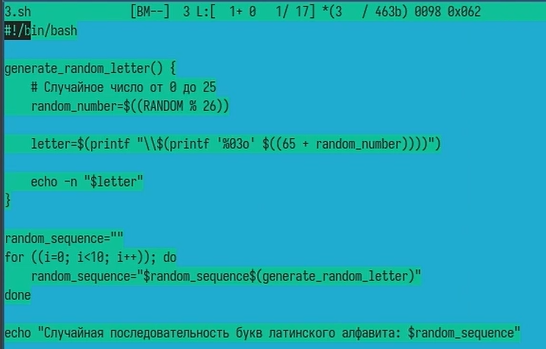
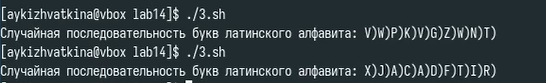


Рис. 10: Программа

Проверяем выполнение программы. (рис. **¿fig:011?**)

{#fig:011width=70%}

# 3 Выводы

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