Отчёт по лабораторной работе №14

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Сергей Витальевич Павлюченков

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, 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.
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 Выполнение лабораторной работы

Написал командный файл, реализующий упрощённый механизм семафоров. Командный файл в течение некоторого времени t1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использует его в течение некоторого времени t2<>t1, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом)

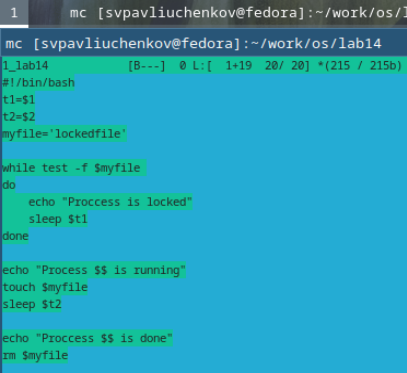


Рис. 1: Код программы семафора

Написал командный файл,который будет запускать код в одном виртуальном терминале в фоновом режиме и перенаправит 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.

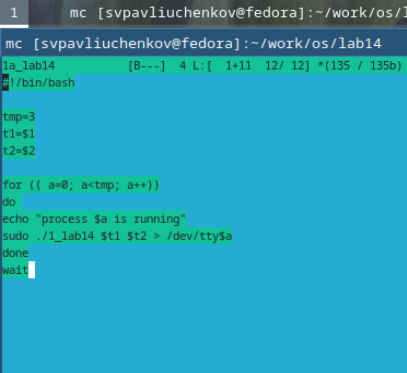


Рис. 2: Код командного файла

Запускаю программу, как видно вывод программы семафора переенаправляется в другой терминал.

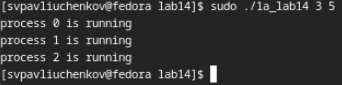


Рис. 3: Запуск программы

Приступаю к выполнению 2-го задания. Изучаю структуру папки man1.

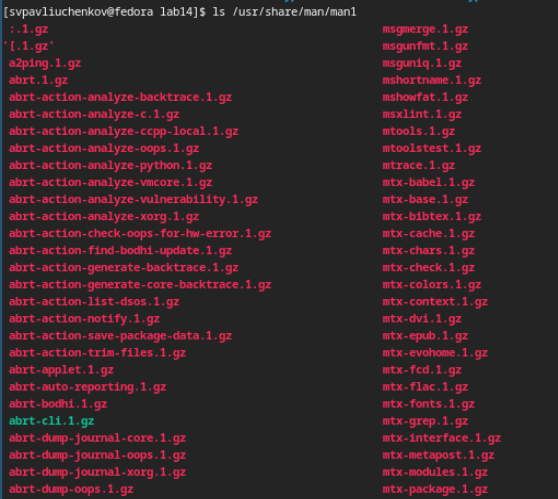


Рис. 4: Страктура папки man1

Реализовал команду man с помощью командного файла. Командный файл получает 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.

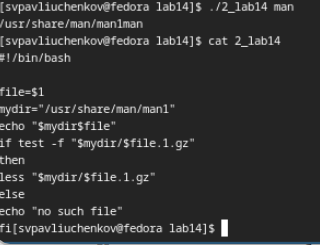


Рис. 5: Код для 2-ой задачи

Открывается командой less сразу же просмотр содержимое справки о команде.

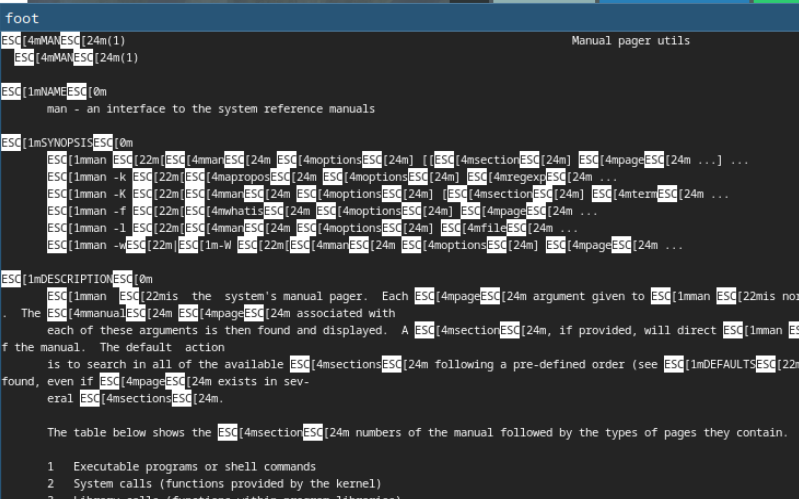


Рис. 6: Запуск программы

Приступаю к 3 задаче. Используя встроенную переменную $RANDOM, написал командный файл, генерирующий случайную последовательность букв латинского алфавита.



Рис. 7: Код для 3-й задачи

Как видно, код генерирует случайную строку из латинских букв.

Запускаю программу # Выводы

Я улучшил свои навыки работы с bash. Изучил команду less, использовал перенаправление в виртуальный терминал и научился пользоваться командой exit().

# 4 Контрольные вопросы

1. Найдите синтаксическую ошибку в следующей строке: while [$1 != “exit”] в том, что данная конструкция скорее всего никогда не закончится, так как $1 не меняется. Еще не хватает пробелов после [ и перед ].
2. Как объединить (конкатенация) несколько строк в одну? a=${$A+$B}
3. Найдите информацию об утилите seq. Какими иными способами можно реализовать её функционал при программировании на bash? например for (( a=1; a<80; a++ )) a будет иметь теже значения, что и seq.
4. Какой результат даст вычисление выражения $((10/3))? Выдаст 3
5. Укажите кратко основные отличия командной оболочки zsh от bash.

ZSH является расширенным аналогом BASH и имеет с ним обратную совместимость, добавляя ему большое количество улучшений. Ключевые особенности ZSH: Встроенное автозаполнение с расширенным функционалом.

1. Проверьте, верен ли синтаксис данной конструкции 1 for ((a=1; a <= LIMIT; a++))

Не хватает пробелов после (( и перед ))

1. Сравните язык bash с какими-либо языками программирования. Какие преимущества у bash по сравнению с ними? Какие недостатки? Bash позволяет напрямую взаимодействовать с консолью, что не позволяет python. Из недостатков