Front matter

lang: ru-RU title: Программирование в командном процессоре ОС UNIX. Расширенное программирование. subtitle: ЛР по ОС №13 author: Танрибергенов Эльдар Марсович group: НПИбд-02-20

Отчёт

о выполнении лабораторной работы № 13.

Программирование в командном процессоре ОС UNIX. Расширенное программирование.

Выполнил:

студент группы НПИбд-02-20

Танрибергенов Эльдар Марсович.

Студ. билет № 1032208074

Москва 2021 г.

Цель работы:

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

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.

Теоретическое введение

Для написания скрипта 1 необходимо выяснить, что такое семафоры. [1]

Семафор – это механизм, который позволяет конкурирующим процессам и потокам работать с общими ресурсами и помогает в решении различных проблем синхронизации таких как гонки, дедлоки (взаимные блокировки) и неправильное поведение потоков.

Для решения этих проблем в ядре присутствуют такие средства как мьютексы, семафоры, сигналы и барьеры.

Есть три вида семафоров:

- Бинарные семафоры (binary semaphore)
- Семафоры-счетчики (counting semaphore)
- Массивы семафоров (semaphore set)

Краткий справочник команд Linux [2]

Ход работы:

Командный файл 1.

https://github.com/emtanribergenov/OS_labs/blob/master/13/screenshots/p1%20(1).png

https://github.com/emtanribergenov/OS_labs/blob/master/13/screenshots/p1%20(2).png

Для испытания подготовил командный файл son, который «спит» 60 секунд.

Файл son.

https://github.com/emtanribergenov/OS_labs/blob/master/13/screenshots/son.png

Испытание:

https://github.com/emtanribergenov/OS_labs/blob/master/13/screenshots/tir1%20(1).png

https://github.com/emtanribergenov/OS_labs/blob/master/13/screenshots/tir1%20(2).png

Командный файл 2.

https://github.com/emtanribergenov/OS_labs/blob/master/13/screenshots/p2.png

Испытание:

https://github.com/emtanribergenov/OS_labs/blob/master/13/screenshots/t2.png

https://github.com/emtanribergenov/OS_labs/blob/master/13/screenshots/tir2%20(1).png

Если данная команда отсутствует в каталоге man1, то выводится соответствующая информация.

https://github.com/emtanribergenov/OS_labs/blob/master/13/screenshots/tir2%20(2).png

Командный файл 3.

https://github.com/emtanribergenov/OS_labs/blob/master/13/screenshots/p3.png

Испытание:

Командный файл отлично сработал -- случайные строки указанного размера сгенерированы и выведены на экран.

https://github.com/emtanribergenov/OS_labs/blob/master/13/screenshots/tir3.png

Вывод:

В результате лабораторной работы я изучил основы программирования в оболочке ОС UNIX, научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Ответы на вопросы:

1. Ошибка: квадратные скобки должны быть отделены пробелом от содержимого.

2. С помощью оператора +=

3. Эта утилита выводит последовательность целых чисел с шагом, заданным пользователем. По-умолчанию, выводимые числа отделяются друг от друга символом перевода строки, однако, с помощью ключа -s может быть задан другой разделитель.

4. Команда bash не найдена.

5. В Zsh оболочками, не входящими в систему, являются .zshrc, а оболочками для входа - .zprofile. Массивы Zsh индексируются от 1 до длины, тогда как Bash индексируется от -1 до длины.

6. Нет.

7.
    - Преимущества: удобство; не нужно устанавливать зависимости.
    - Недостатки: не знаю.