Лабораторная работа №12

Курс “Операционные Системы”

Лихтенштейн А.А., НКАбд-03-22

29 апреля 2023

Российский университет дружбы народов, Москва, Россия

Докладчик

- Лихтенштейн Алина Алексеевна
- студент группы НКАбд-03-22
- кафедры Компьютерные и информационные науки
- Российский университет дружбы народов
- 1132229533@pfur.ru

Цель работы

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

Задачи

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.

Задачи

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.

Задачи

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.

Выполнение лабораторной работы

Исходный код скрипта №1

semaphore.sh - GNU Emacs at aaliechtenstein

File Edit Options Buffers Tools Sh-Script Help

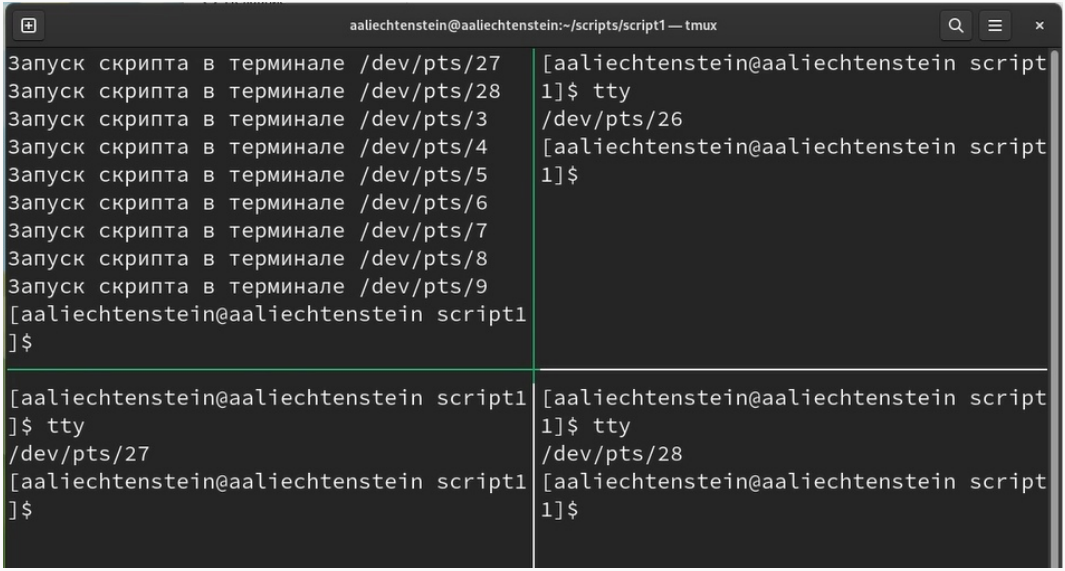
Save Undo

```
#!/bin/bash

resource="semaphore.txt"
t1=5
t2=10

# установка блокировки на ресурсе
exec 200>"$resource"
flock -x 200

# дожидаемся освобождения ресурса
echo "Waiting for resource..."
sleep $t1

# использование ресурса
echo "Using resource..."
echo "Process $$ is using the resource" > "$resource"
sleep $t2

# освобождение ресурса
flock -u 200
echo "Resource released"
```

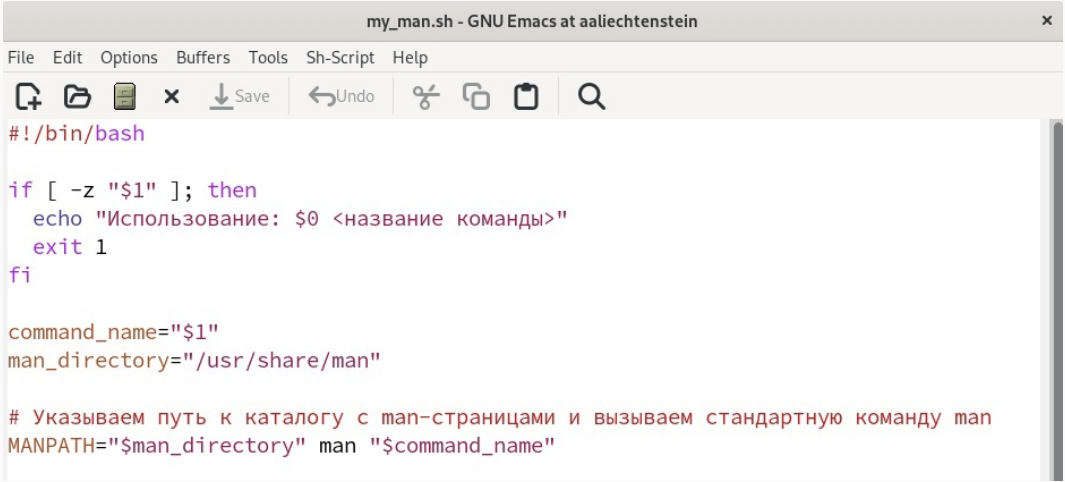
Описание скрипта №1

Данный скрипт реализует упрощенный механизм семафоров на основе файловой блокировки. Он дожидается освобождения ресурса, выдавая об этом сообщение, затем использует его, выдавая информацию о том, что ресурс используется соответствующим процессом. Ресурс защищен блокировкой на файле semaphore.txt. Скрипт можно запустить в фоновом режиме и перенаправить его вывод в другой терминал.

Результат выполнения скрипта №1



Исходный код скрипта №2



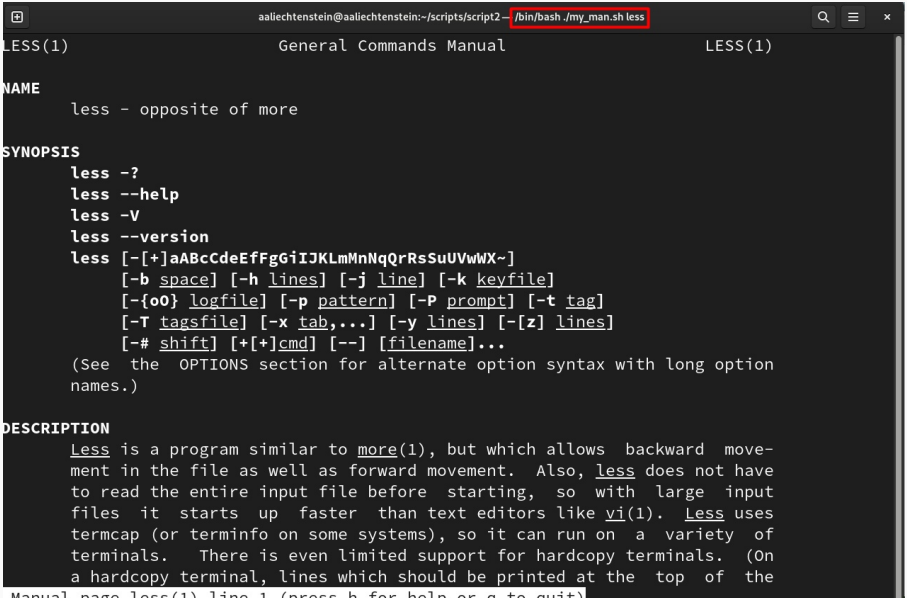

```
#!/bin/bash

if [ -z "$1" ]; then
  echo "Использование: $0 <название команды>"
  exit 1
fi

command_name="$1"
man_directory="/usr/share/man"

# Указываем путь к каталогу с man-страницами и вызываем стандартную команду man
MANPATH="$man_directory" man "$command_name"
```

Описание скрипта №2

Этот скрипт предоставляет пользователю возможность получить справку для команды, указанной в качестве аргумента. Он использует стандартную команду man с указанием пути к каталогу с man-страницами через переменную окружения MANPATH.

Скрипт проверяет, указано ли имя команды в качестве аргумента. Если аргумент не указан, выводит справочное сообщение и завершает работу. Запоминает указанное имя команды и путь к каталогу man-страниц. Устанавливает значение переменной окружения MANPATH равным пути к каталогу man-страниц. Вызывает стандартную команду man с именем команды в качестве аргумента.

Результат выполнения скрипта №2



Исходный код скрипта №3

script.sh - GNU Emacs at aaliechtenstein

File Edit Options Buffers Tools Sh-Script Help

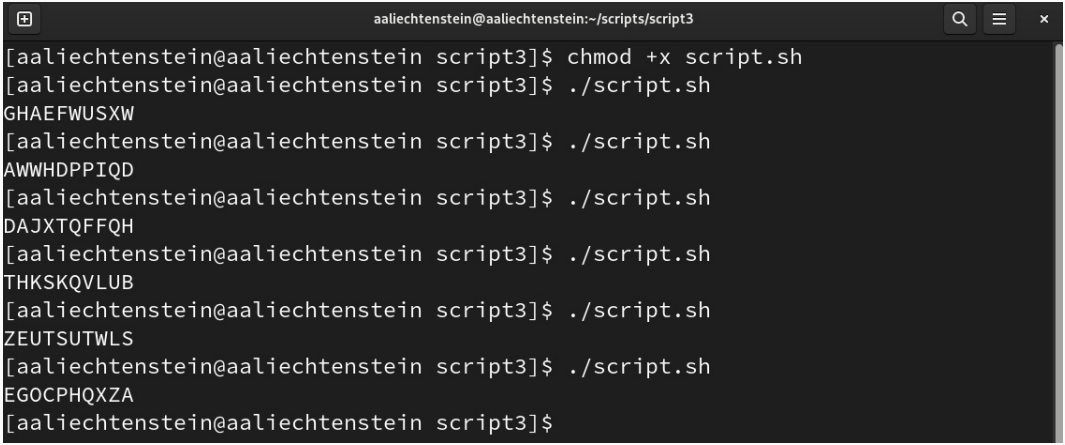
Save Undo

```
#!/bin/bash

# Задайте количество букв в последовательности
sequence_length=10

# Создайте пустую строку
random_string=""

# Цикл для генерации последовательности
for ((i=0; i<$sequence_length; i++)); do
  # Генерируйте случайное число от 0 до 25 с использованием $RANDOM
  random_number=$(($RANDOM % 26))

  # Преобразуйте случайное число в соответствующую букву латинского алфавита
  random_letter=$(printf "\\$(printf '%03o' $((random_number + 65)))")

  # Добавьте случайную букву к строке
  random_string+="$random_letter"
done

# Выведите случайную последовательность
echo "$random_string"
```

Описание скрипта №3

Скрипт генерирует случайную последовательность латинских букв заданной длины. Он использует переменную $RANDOM для получения псевдослучайных чисел, преобразует их в соответствующие буквы и объединяет их в строку. Затем выводит эту строку на экран.

Результат выполнения скрипта №3



Результаты

Выводы

Были получены практические навыки написания более сложные командных файлов с использованием логических управляющих конструкций и циклов.