Отчет по файлам Лабораторной работы 14

# Шилов Александр Ильич

11322446746

Этот отчет предоставляет обзор скриптов оболочки и соответствующих изображений вывода, найденных в каталоге лабораторной работы.

# my\_man.sh

/bin/bash

if [ $# -eq е ] ; then echo ”Used $6 «ommand>” exit 1

cmd=” $1 ” man\_page=” /usr/share/man/man1/${cmd} . 1. gz ”

if ”$man\_page” ] ; then zcat ”$man\_page” 1ess

echo ”Command isnt found ' $cmd '

Назначение: Этот скрипт представляет собой упрощенный просмотрщик страниц руководства. Он принимает имя команды в качестве аргумента и отображает ее тап-страницу, если она существует.

Функциональность:

* Проверяет, предоставлен ли аргумент команды
* Ищет страницу руководства в стандартном месте (/usr/share/man/man1/)
* Если найдено, использует zcat для распаковки и 1ess для отображения страницы руководства
* В противном случае отображает сообщение об ошибке

2. random letters.sh

/bin/bash

-16}



RANDOM % 26 ) ) printf ”\\x$(printf 7092х $((65 + rand)))”

resu1t= fop ((i=e; i<1ength; i++)); do



done

echo $pesu1t

Назначение: Генерирует случайную строку из заглавных букв.

Функциональность:

* Принимает необязательный аргумент для длины строки (по умолчанию 10)
* Использует функцию для генерации случайных заглавных букв ASCII
* Объединяет буквы в результирующую строку
* Выводит случайную строку

На изображении показано, как скрипт генерирует случайные заглавные буквы.

Скрипт можно использовать с аргументом для указания длины случайной строки.

З. samaphore.sh

/bin/bash

## # variables

t-1=5 t-2=3 sem\_file=" /tmp/ semaphore . lock"

# our function wait for\_semaphore() { echo "Wainting for the resource. . . " while [ -f "$sem\_file" ] && [ $tl -gt e do sleep 1

( (tl- -) ) done if [ ! \_ f "$sem\_file" ] ; then touch "$sem\_file" echo "Process captured resouces" return 

echo Couldnt capture the resouces" exit 1

# capturing resources wait for\_semaphore

# usinng the resources echo "Used resource ( $$ ) in $t2 sec. . " sleep $t2

# freeing resourcces rm "$sem\_file" echo "Resourcees freed $$"

Ha3HaqeHne: Aerv10HcTpvqpyeT 6a30BY10 PeanV13al.4V11-O cervlacþopa B bash AJ1fi ynpaBneHl,1fl pecypcaML,1.

OYHKL4VIOHaJ1bHOCTb:

* L'1cn0J1b3yeT (þaVlJ1 6JlOKVIPOBKV1 (/tmp/semaphore.lock) B KaqecTBe cervlacþopa
* Реализует механизм ожидания с таймаутом (5 секунд)
* Когда процесс получает ресурс, он создает файл блокировки
* Процесс использует ресурс в течение указанного времени (З секунды)
* После использования ресурса удаляет файл блокировки для освобождения ресурса
* Включает Ю процесса в выходные сообщения для идентификации

На изображении показано выполнение скрипта семафора, демонстрирующее, как процессы ожидают и получают ресурсы с использованием механизма семафора.

# Дополнительный контент

Также есть другие РЖЕ-файлы, которые показывают различные запуски или аспекты скриптов:

1 . contentrandom.png - Показывает дополнительный вывод из скрипта случайных букв

2. samapohore.png - Показывает еще одно выполнение скрипта семафора

# Резюме

Эта лабораторная работа, по-видимому, сосредоточена на написании скриптов bash с тремя основными примерами:

1. Пользовательский просмотрщик страниц руководства
2. Генератор случайных букв

З. Базовая реализация семафора для управления ресурсами

Каждый скрипт демонстрирует различные концепции программирования в bash:

* Обработка файлов и выполнение команд (my\_man.sh)
* Генерация случайных чисел и манипуляции со строками (random letters.sh)
* Синхронизация процессов и управление ресурсами (samaphore.sh)