Отчёт по лабораторной работе №13

Операционные системы

Ильина Любовь Александровна

Содержание

# Цель работы

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

# Выполнение лабораторной работы

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. (рис. [1](#fig:001) - [3](#fig:003))



Figure 1: Командный файл

Figure 2: Запуск вывода файла в текстовую консоль

Figure 2: Запуск вывода файла в текстовую консоль

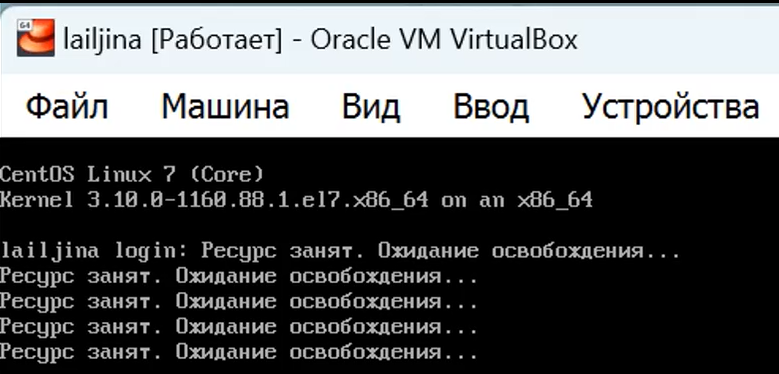


Figure 3: Исполнение командного файла в текстовой консоли

1. 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. (рис. [4](#fig:004) - [5](#fig:005))

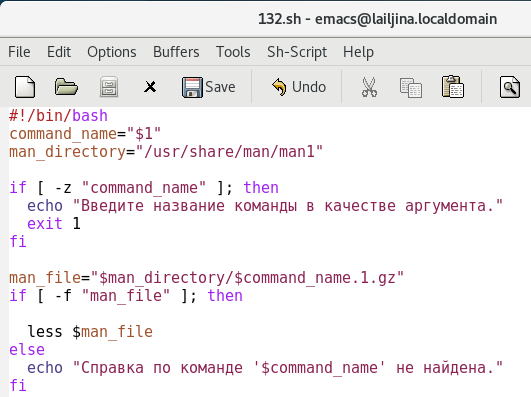


Figure 4: Командный файл



Figure 5: Исполнение командного файла

1. 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 (рис. [6](#fig:006) - [7](#fig:007))

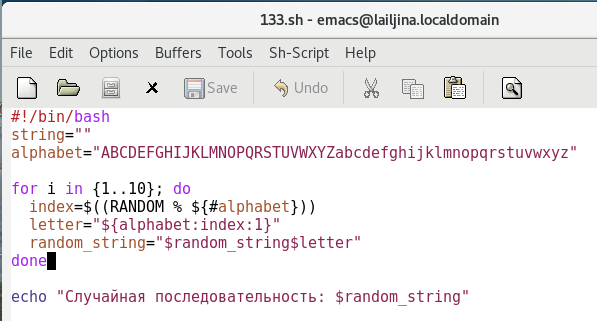


Figure 6: Командный файл

Figure 7: Исполнение командного файла

Figure 7: Исполнение командного файла

1. Исправление синтаксической ошибки в строке while [$1 != “exit”] : while [[ $1 != “exit” ]]
2. Как объединить (конкатенация) несколько строк в одну? с помощью + и = можно объединить несколько строк в одну:concatenated\_string=“string2”
3. Найдите информацию об утилите seq. Какими иными способами можно реализовать её функционал при программировании на bash? Утилита seq в bash используется для генерации последовательностей чисел. Реализовать функционал возможно циклами:

for ((i=1; i<=10; i++))  
current=$start  
while [ $current -le $end ]  
do  
 echo "$current"  
 current=$((current + 1))  
done

1. Какой результат даст вычисление выражения $((10/3))? целочисленное деление даст результат 3.
2. Укажите кратко основные отличия командной оболочки zsh от bash. Конфигурационные файлы: Bash использует файлы ~/.bashrc и ~/.bash\_profile для настройки окружения, а Zsh использует файлы ~/.zshrc и ~/.zprofile. У Zsh также есть более сложная и гибкая система настройки, позволяющая использовать файлы конфигурации, такие как ~/.zshrc, ~/.zshenv, ~/.zlogin и другие.

Автодополнение: Zsh обладает мощным и гибким механизмом автодополнения, который предлагает подсказки и автодополнение команд, параметров и файлов. Bash также имеет функциональность автодополнения, но не такую мощную и настраиваемую, как в Zsh.

Синтаксис: Zsh имеет некоторые отличия в синтаксисе и некоторые дополнительные возможности по сравнению с Bash. Например, в Zsh поддерживается прямая подстановка переменных без использования кавычек, и существуют другие изменения в работе с параметрами командной строки.

1. Проверьте, верен ли синтаксис данной конструкции for ((a=1; a <= LIMIT; a++)) верно
2. Сравните язык bash с какими-либо языками программирования. Какие преимущества у bash по сравнению с ними? Какие недостатки? Bash является командной оболочкой и языком сценариев (скриптовым языком), предназначенным для автоматизации задач в операционной системе Unix и Unix-подобных системах. Вот сравнение bash с Python: Преимущества bash:
3. Легко интегрируется с системными командами и утилитами.
4. Отлично подходит для выполнения системных задач и автоматизации.
5. Простой и быстрый для написания скриптов. Недостатки bash:
6. Ограниченные возможности для сложных алгоритмов и структур данных.
7. Отсутствие некоторых функций и библиотек, которые могут быть доступны в других языках, таких как Python.

# Выводы

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