|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №8**

**«Создание сценария для конфигурации системы»**

**ДИСЦИПЛИНА: «Операционные системы»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-62Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Красавин Е.В. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2023

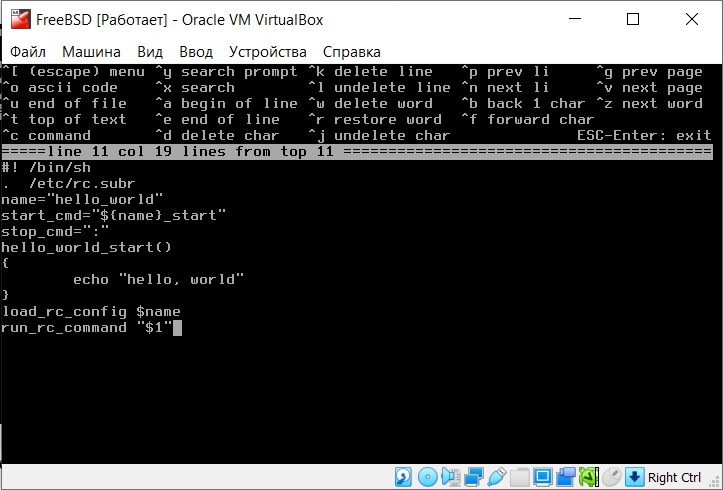
**Цель:** закрепление полученных навыков по настройке основных сервисов системы FreeBSD.

**Задачи:** сконфигурировать систему исходя из заданной вам схемы сети (сетевые интерфейсы, маршрутизация, DNS).

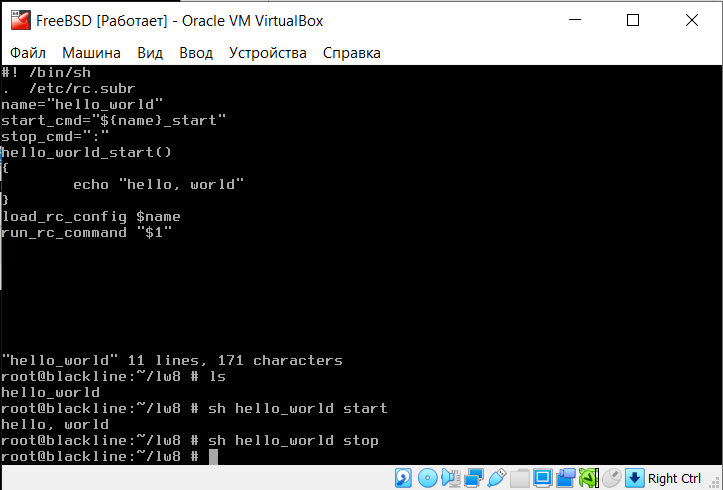
**Задание:**

1. Ознакомиться с предложенным материалом для получения информации об управлении учетными записями в ОС FreeBSD
2. Создать простейший сценарий.
3. Настроить простейший сценарий.
4. Создать простейший демон
5. Запустить простейший демон.
6. Создать более сложный демон.
7. Присоединить сценарий к инфраструктуре.
8. Написать сценарий исходя из заданной схемы сети.
9. Проверить работоспособность написанного скрипта.
10. Сделать скрипт более гибким.

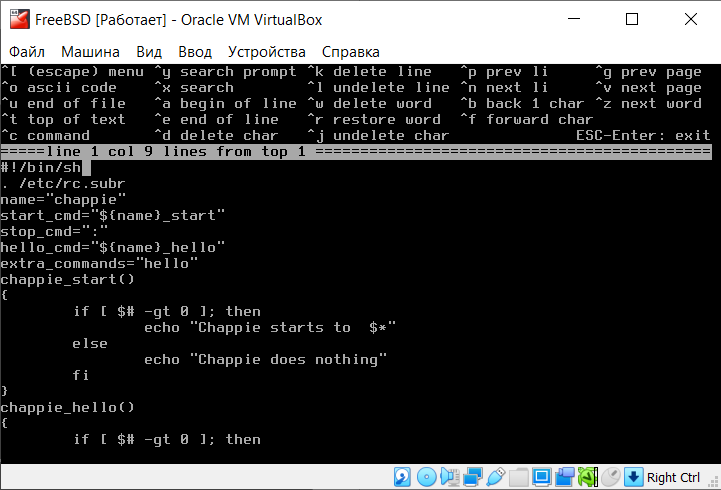
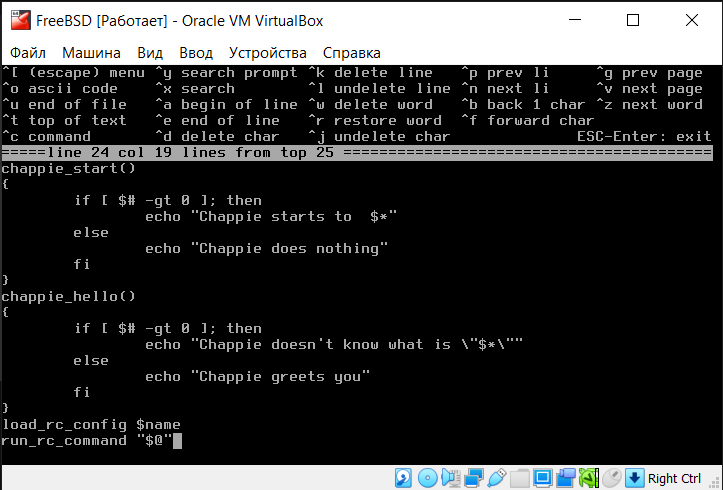
**Результат:**

****

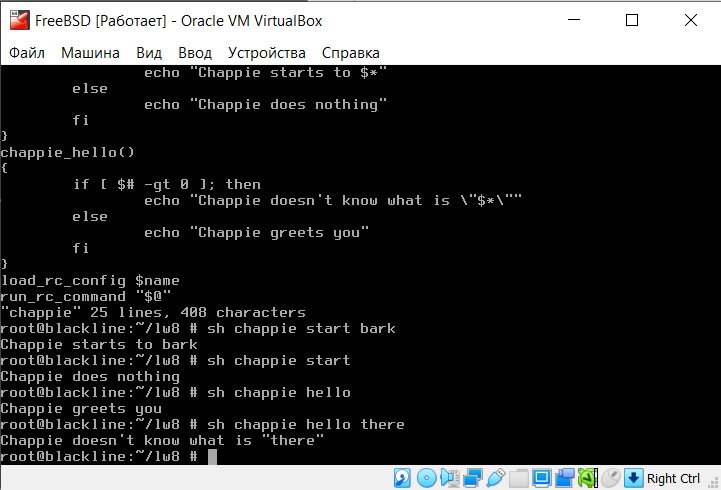
**Рис. 1.** Простейший сценарий

****

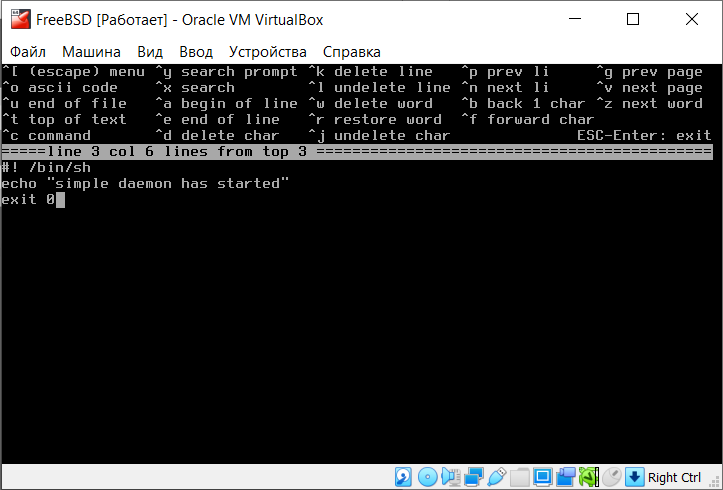
**Рис. 2.** Работа сценария

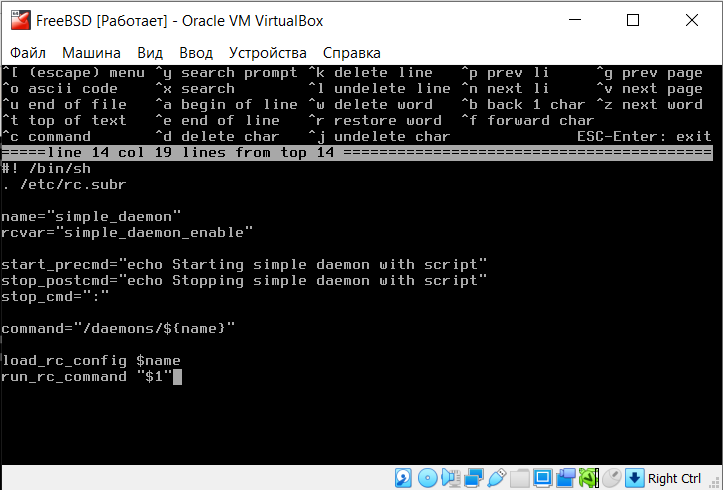
**Рис. 3.** Более гибкий сценарий

****

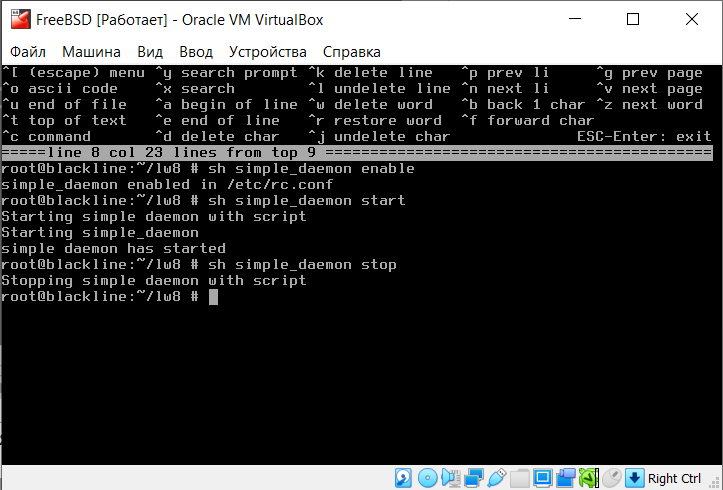
**Рис. 4.** Работа более гибкого сценария



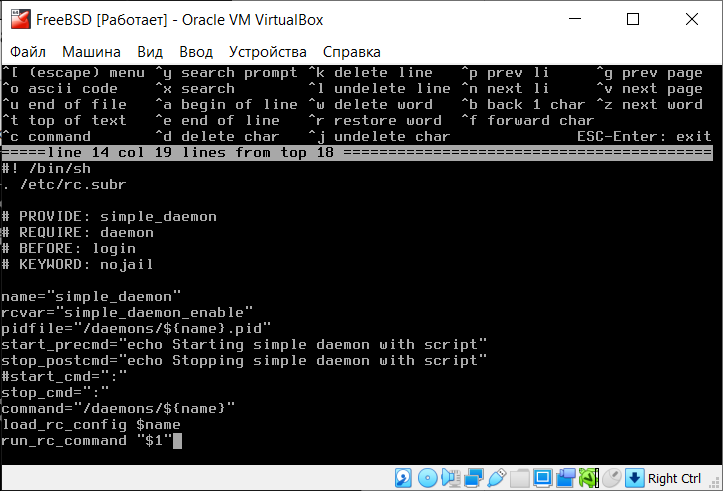
**Рис. 5.** Простой демон

****

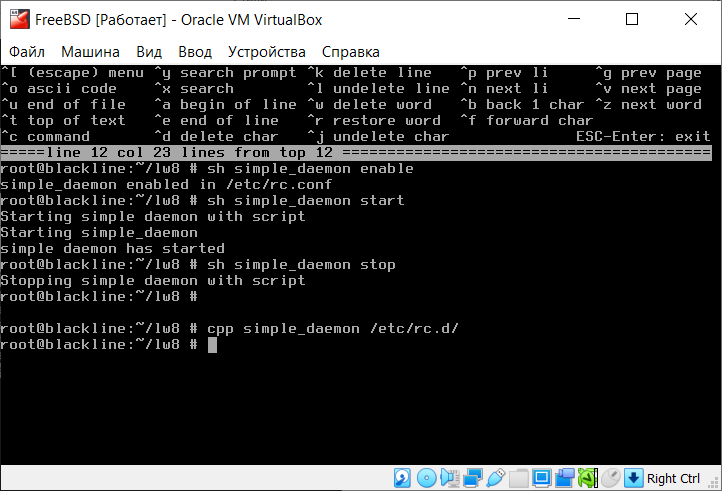
**Рис. 6.** Сценарий для управления демоном



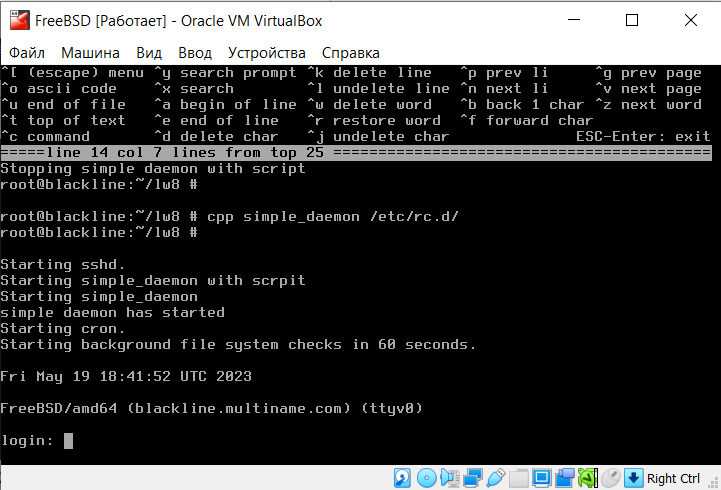
**Рис. 7.** Работа демона



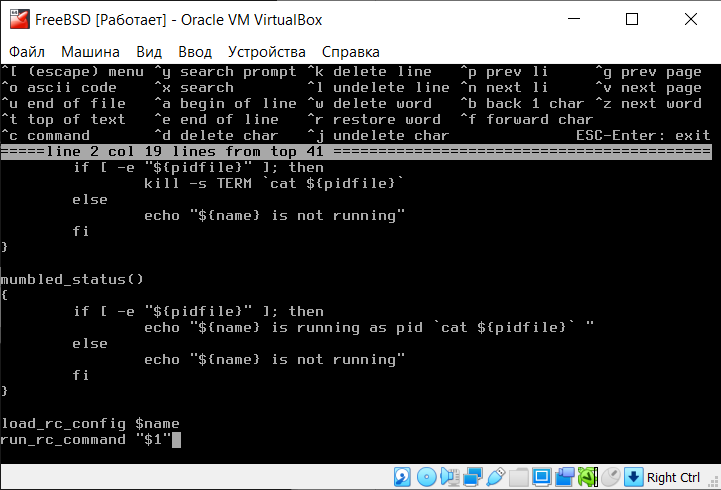
**Рис. 8.** Изменение сценария



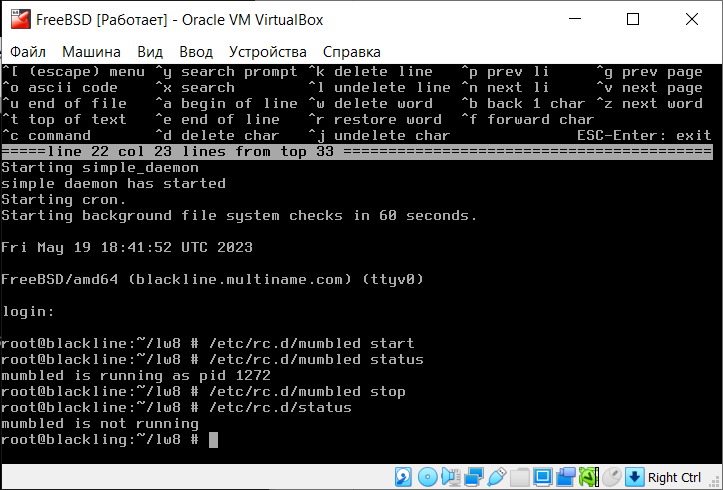
**Рис. 9.** Копирование сценария



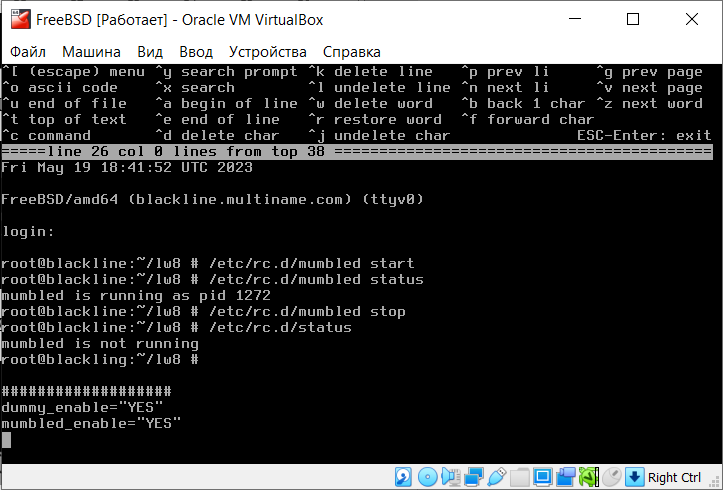
**Рис. 10.** Запуск сценария вместе с системой



**Рис. 11.** Сложный демон



**Рис. 12.** Работа сложного демона



**Рис. 13.** Добавление сценариев к инфраструктуре

**Вывод:** в ходе выполнения лабораторной работы были закреплены полученные навыки по настройке основных сервисов системы FreeBSD.

**Контрольные вопросы:**

1. **Объясните, что такое rc.d.**

Это модульный скрипт запуска системы.

1. **Объясните, что такое сценарий.**

Скрипт – это понятие в программировании, обозначающее последовательность команд для выполнения конкретных операций. По сути, это небольшая программа, заточенная под определенное действие.

1. **Объясните, что такое демон.**

Демон — компьютерная программа в UNIX-подобных системах, запускаемая самой системой и работающая в фоновом режиме без прямого взаимодействия с пользователем.

1. **Раскройте область применения демонов.**

Демоны в Unix-системах выполняют ряд важных функций, включая:

* Сетевые службы: Демоны могут предоставлять сетевые службы, такие как веб-серверы, FTP-серверы, DNS-серверы и т.д. Эти службы работают в фоновом режиме и готовы к обработке запросов от клиентов в любое время.
* Резервное копирование: Демоны также используются для резервного копирования файлов и системных данных. Например, демон cron запускает задачи резервного копирования в заданные промежутки времени.
* Мониторинг: Демоны могут быть настроены для мониторинга системных ресурсов, таких как использование процессора, памяти и дискового пространства. Они могут предупреждать администратора системы, если ресурсы исчерпываются или если происходят ошибки.
* Системный журнал: Демон syslog отвечает за запись системных журналов, в которых хранятся сообщения об ошибках, событиях и других важных сведениях. С помощью этих журналов можно отслеживать работу системы и быстро реагировать на проблемы.
* Автоматическая настройка: Демоны также могут использоваться для автоматической настройки системы и ее компонентов. Например, демон udev отвечает за автоматическое определение и настройку новых устройств, подключенных к системе.

1. **Перечислите команды для работы со сценариями.**

chmod, echo, if и т.д.

1. **Раскройте суть аргументов сценария.**

Аргументы в сценарии - это значения, переданные скрипту при его запуске. Аргументы обычно представлены в виде строк и могут быть использованы внутри сценария для выполнения различных задач.

Аргументы передаются скрипту через командную строку. Каждый аргумент разделяется пробелом и может содержать любые символы, включая пробелы, кавычки и специальные символы.

1. **Опишите назначение sh.**

sh - это командный интерпретатор для Unix-подобных систем, который используется для выполнения командных скриптов. Он является стандартным оболочкой командной строки для большинства Unix-подобных операционных систем.

1. **Раскройте смысл PID.**

PID - это сокращение от "Process Identifier" (идентификатор процесса). Это уникальный числовой идентификатор, который операционная система присваивает каждому процессу, запущенному на компьютере.

PID используется для идентификации процесса в системе, например, чтобы управлять им или завершить его выполнение. Когда процесс запускается, операционная система присваивает ему уникальный PID. PID может быть использован для определения статуса процесса, в том числе для определения, работает ли процесс, завершился ли он, или возникли ли какие-либо проблемы во время его выполнения.

1. **Объясните, зачем нужен Makefile.**

Makefile - это текстовый файл, который содержит инструкции для автоматической компиляции, сборки и установки программного обеспечения. Он используется для автоматизации процесса сборки программного обеспечения и упрощения процесса разработки.

В Makefile содержатся правила для компиляции и сборки исходного кода, а также для генерации различных файлов, таких как документация, исполняемые файлы и т.д. При запуске команды make в командной строке, он автоматически обрабатывает Makefile и выполняет необходимые действия для сборки программы или проекта.

1. **Перечислите преимущества гибких скриптов.**

Гибкие скрипты имеют несколько преимуществ, включая:

* Автоматизация повторяющихся задач: Гибкие скрипты помогают автоматизировать повторяющиеся задачи, что позволяет сэкономить время и уменьшить вероятность ошибок.
* Удобство и простота использования: Гибкие скрипты легки в использовании и не требуют значительных знаний программирования. Они позволяют легко настраивать их поведение и обеспечивают возможность быстро и легко выполнять задачи.
* Адаптируемость: Гибкие скрипты позволяют быстро и легко адаптировать поведение программы к изменяющимся условиям и требованиям.
* Переносимость: Гибкие скрипты могут выполняться на различных платформах и операционных системах, что обеспечивает удобство использования в различных средах.
* Возможность интеграции: Гибкие скрипты могут использоваться для интеграции различных программных систем и утилит, что позволяет упростить и ускорить процесс работы.