**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Кафедра “фундаментальная информатика и информационные технологии”**

**отчет**

**по домашнему заданию “ Создание демона для регулярного резервного копирования данных ”**

**по дисциплине «Системное программирование в Linux»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 22Б16 |  | Олизько С.С. |
| Преподаватель |  | Киямов Ж.У. |

**Санкт-Петербург**

**2023 г.**

**Оглавление**

1. Цель работы
2. Задача
3. Теоритическая часть
4. Алгоритм метода
5. Описание программы
6. Рекомендации по использованию
7. Вывод

**Цель:**

Разработать и внедрить демона на языке C++, обеспечивающего автоматическое регулярное резервное копирование данных из исходного каталога в другой.

**Задачи:**

Написать код демона на языке C++, который в фоновом режиме ожидает наступления времени для следующей резервной копии. Создает резервные копии файлов из исходного каталога. Внедрить демона в систему так, чтобы он автоматически запускался при старте операционной системы. Обеспечить регулярное выполнение резервного копирования в соответствии с настройками. Создать механизм управления демоном через командную строку.

**Теоретическая часть:**

`systemctl` представляет собой ключевой инструмент в системе `systemd` для управления службами и компонентами операционной системы в современных дистрибутивах Linux. Этот утилитарный инструмент обеспечивает единый интерфейс для контроля за различными аспектами системы, включая службы, сокеты, устройства и другие ресурсы. С помощью команд `systemctl` можно легко запускать, останавливать, перезапускать и мониторить состояние системных компонентов. Он также предоставляет функционал для настройки автозапуска служб при загрузке системы, а также для работы с журналом системных событий.

PID (Process ID) - это уникальный идентификатор процесса в операционной системе. Каждый запущенный процесс получает свой собственный PID, который является положительным целым числом. Значение PID может использоваться для идентификации и управления процессами в системе.

Демоны - это фоновые процессы в операционной системе, которые выполняются в фоновом режиме и не привязаны к конкретному пользовательскому сеансу. Они обеспечивают выполнение различных системных задач, таких как обслуживание оборудования, работа с сетью или выполнение периодических задач.

**Алгоритм метода:**

1. Подключение необходимых библиотек для работы с потоками, файлами, логами и сигналами.
2. Определение глобальной переменной is\_running, отвечающей за состояние работы демона.
3. Задание пути к конфигурационному файлу в configPath.
4. Открывает системный лог.
5. Устанавливает обработчики сигналов.
6. Выводит сообщение о запуске демона.
7. Вызывает основной цикл демона (daemonLoop).
8. Закрывает системный лог при завершении работы.

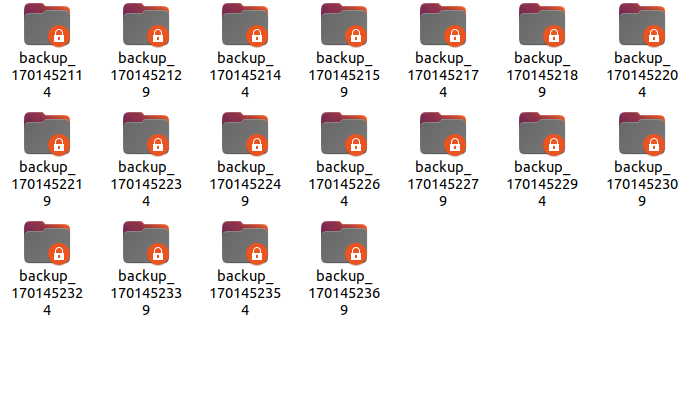
**Описание программы:**

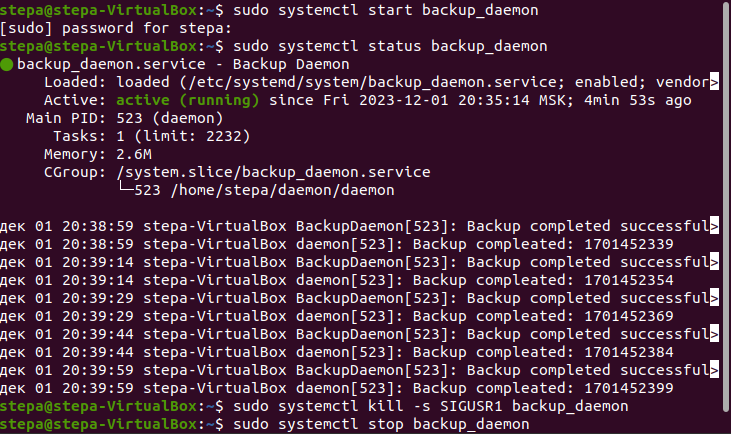
|  |  |  |
| --- | --- | --- |
| **Имя функции** | **Тип возвращаемого значения** | **Описание функции** |
| readConfig | bool | Читает конфигурационные данные из файла и заполняет структуру Config. Возвращает true при успешном чтении, иначе false. |
| backupData | void | Выполняет резервное копирование данных из указанной директории в другую, добавляя временной штамп к имени папки. Логирует результат и выводит информацию в консоль. |
| daemonLoop | void | Основной цикл демона, который читает конфигурацию, выполняет резервное копирование и ожидает следующего выполнения в соответствии с интервалом. |
| pause\_handler | void | Обработчик сигнала SIGTSTP, устанавливающий флаг is\_running в false и логирующий сообщение о паузе. |
| continue\_handler | void | Обработчик сигнала SIGCONT, устанавливающий флаг is\_running в true и логирующий сообщение о продолжении работы. |
| my\_terminate\_handler | void | Обработчик сигнала SIGTERM, выполняющий завершение демона, закрывающий системный лог и выходящий с кодом завершения EXIT\_SUCCESS. |
| status\_handler | void | Обработчик сигнала SIGUSR1, выводящий в системный лог информацию о текущем состоянии демона (работает или приостановлен). |
| setup\_signal\_handlers | void | Устанавливает обработчики сигналов для демона. |
| main | int | Точка входа в программу, открывающая системный лог, устанавливающая обработчики сигналов, выводящая сообщение о запуске демона и вызывающая основной цикл демона. |

**Рекомендации по использованию:**

1. **Конфигурационный файл:**
   1. Убедитесь, что путь к конфигурационному файлу backup\_config.conf указан корректно в переменной configPath в исходном коде программы.
2. **Настройка конфигурации:**
   1. Редактируйте файл backup\_config.conf, чтобы задать значения для source\_directory (исходная директория), backup\_directory (директория для резервных копий) и backup\_interval (интервал между резервными копиями в секундах).
3. **Запуск программы:**
   1. Соберите программу и запустите исполняемый файл.
   2. При запуске, программа будет начинать выполнение резервного копирования с указанными в конфигурации параметрами.
4. **Управление работой демона:**
   1. Демон может быть приостановлен, продолжен, а также завершен по командам с использованием сигналов. Используйте следующие сигналы:
      1. SIGTSTP для паузы (передать сигнал при помощи команды kill -SIGTSTP <pid>),
      2. SIGCONT для продолжения (передать сигнал при помощи команды kill -SIGCONT <pid>),
      3. SIGTERM для завершения (передать сигнал при помощи команды kill -SIGTERM <pid>),
      4. SIGUSR1 для получения статуса (передать сигнал при помощи команды kill -SIGUSR1 <pid>).
5. **Логирование:**
   1. Результаты выполнения программы логируются в системный журнал (syslog). Вы можете проверять системный журнал для получения информации о ходе выполнения.
6. **Обработка ошибок:**
   1. В случае возникновения ошибок при чтении конфигурации или выполнении резервного копирования, информация об ошибках будет записана в системный лог.

**Контрольный пример:**



****

**Вывод:**

Программа резервного копирования демона разработана для автоматического выполнения резервного копирования данных в системе с заданным интервалом.

**Список литературы:**

<https://en.wikipedia.org/wiki/Daemon_(computing)>

<https://itsfoss.com/linux-daemons/>

<https://www.digitalocean.com/community/tutorials/how-to-use-systemctl-to-manage-systemd-services-and-units-ru>

**GitHub:**

https://github.com/StephanOlizko/backup\_daemon