**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

**“Большие данные и распределенная цифровая платформа”**

**ОТЧЁТ**

**По лабораторной работе №2 По дисциплине “Системное программирование в линукс”**

**На тему “Создание демона для регулярного резервного копирования данных”**

**Студент гр. 23Б16-пу**

**Пушкарев Н.П.**

**Преподаватель**

**Киямов Ж.У.**

**Санкт-Петербург**

**2024 г.**

[Цель работы 3](#__RefHeading___Toc240_2625492298)

[Описание задачи 3](#__RefHeading___Toc256_2625492298)

[Теоретическая часть 3](#__RefHeading___Toc254_2625492298)

[Описание программы 4](#__RefHeading___Toc252_2625492298)

[Основные шаги программы 4](#__RefHeading___Toc250_2625492298)

[Рекомендации пользователя 5](#__RefHeading___Toc248_2625492298)

[Рекомендации программиста 5](#__RefHeading___Toc246_2625492298)

[Контрольный пример 5](#__RefHeading___Toc244_2625492298)

[Вывод 7](#__RefHeading___Toc242_2625492298)

[Список источников 8](#__RefHeading___Toc260_2625492298)

# Цель работы

Цель данной лабораторной работы — разработка системного демона на языке C++ для автоматического резервного копирования данных с одного каталога в другой с регулярной периодичностью. Демон должен поддерживать загрузку конфигурации из файла, выполнять резервное копирование с временной меткой, вести системный журнал событий и автоматически запускаться при старте операционной системы.

# Описание задачи

Задача заключается в создании и настройке системного демона для автоматического резервного копирования данных. Программа должна обеспечивать:

Чтение конфигурации из файла, содержащего исходный и целевой каталоги, а также частоту выполнения.

Автоматический запуск в фоновом режиме с ожиданием установленного интервала времени.

Копирование данных с добавлением временной метки к каждой резервной копии.

Журналирование всех операций.

Интеграцию с systemd для автозапуска при старте системы.

# Теоретическая часть

Для решения задачи используется язык программирования C++. Основные аспекты включают:

Создание и использование файлов конфигурации для гибкой настройки параметров работы.

Создание процесса-демона, работающего в фоновом режиме.

Обработка сигналов для безопасного завершения работы демона.

Ведение системного журнала через syslog.

Интеграция с systemd для автоматического управления процессом.

# Описание программы

Программа написана на C++ и разделена на модули, выполняющие отдельные функции:

исходный код: backupd.cpp

Таблица 1. Описание исходного кода

|  |  |  |
| --- | --- | --- |
| Название функции | Входные данные | Описание |
| class Config |  | класс для чтения и загрузки параметров из конфигурационного файла backup\_config.ini |
| Config.loadconfig | (const std::string& filename)  На вход получает абсолютный путь конфигурационного файла | функция класса Config, предназначенная для чтения конфигурационного файла |
| signalHandler | (int signum)  На вход получает сигнал | обработчик сигналов для безопасного завершения работы демона при получении SIGTERM или SIGINT. |
| backupFiles | (const std::string& sourceDir, const std::string& backupDir)  На вход получает абсолютный путь директорий: источника и бэкапа. | функция, копирующая файлы с временной меткой в целевой каталог. |
| getCurrentTimestamp | () | получает текущее время системы и выдает строку в виде %Y%m%d\_%H%M%S |
| main | () | функция, запускающая демона в фоновом режиме и управляющая его выполнением в цикле. |

# 

# Основные шаги программы

1. Чтение конфигурации из backup\_config.ini, содержащей sourceDir, backupDir и backupFrequency.
2. Переход демона в фоновый режим.
3. Запуск цикла, который:
   1. Ожидает установленный интервал времени.
   2. Копирует файлы с временной меткой.
   3. Записывает данные о копировании в системный журнал.
4. Остановка демона при получении сигнала.

# Рекомендации пользователя

1. Убедитесь, что файл backup\_config.ini содержит правильные пути к исходной и целевой директориям и указан нужный интервал выполнения.
2. Для запуска программы:
   1. Скомпилируйте демона командой g++ backupd.cpp -o backupd -std=c++17.
   2. Скопируйте скомпилированный файл в директорию, указанную в backupd.service.
3. Для интеграции демона в систему используйте systemctl:
   1. Запустите sudo systemctl enable backupd.service для автозапуска при старте системы.
   2. Запустите и остановите службу командами sudo systemctl start backupd.service и sudo systemctl stop backupd.service

# Рекомендации программиста

* Следите за актуальностью используемых библиотек, поддерживающих работу с файлами и журналированием.
* Оптимизируйте использование ресурсов, добавляя проверки на доступность файлов и правильность путей.
* Добавьте обработку исключений для более детальной диагностики ошибок при загрузке конфигурации и копировании файлов.

# Контрольный пример

Пример файла backup\_config.ini с параметрами:

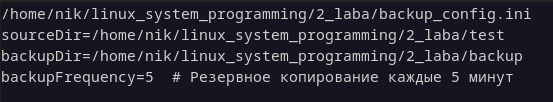
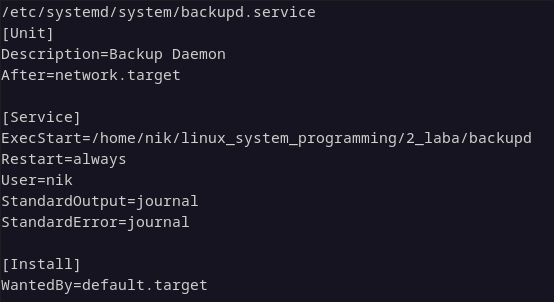
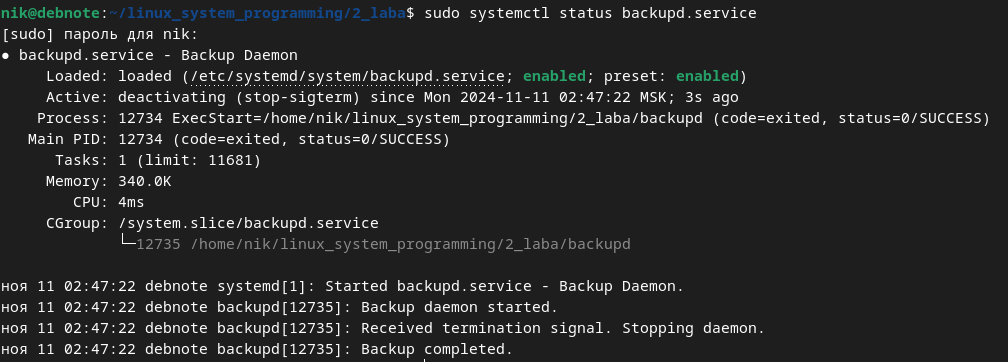


Рис1. Пример backup\_config.ini

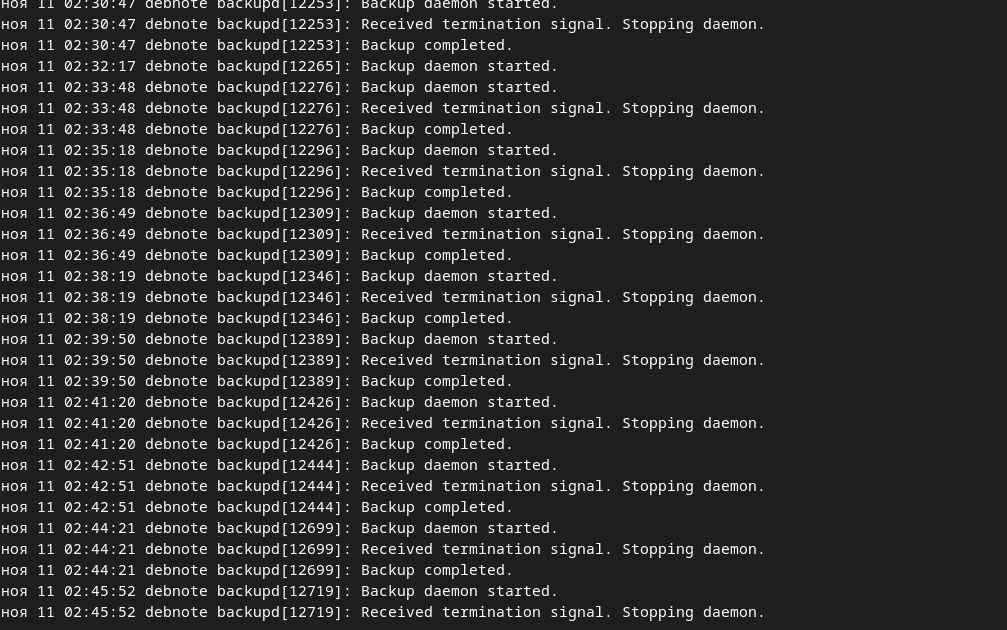
Пример backupd.service - конфигурационного файла, предназначенного для запуска и управления системным демоном с помощью systemd в операционной системе Linux

Рис2. Пример backupd.service

Пример статуса работы демона.

Рис3. Пример статуса работы демона

Проверьте лог в системном журнале через journalctl -u backupd.service.

Ри4. Пример логов

# Вывод

В ходе лабораторной работы был разработан демон на C++ для автоматического резервного копирования данных. Демон был успешно интегрирован с системой через systemd, обеспечивает регулярное выполнение задач, а также поддерживает системное журналирование и конфигурацию через файл backup\_config.ini. Работа продемонстрировала основные аспекты системного программирования, такие как управление процессами, работа с файловой системой и интеграция с systemd.

# Список источников

* Linux Manual Pages: man 7 daemon – Основы создания и управления демонами в Linux
* Интернет-источник: https://www.altlinux.org/%D0%A1%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5\_%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81%D0%B0\_systemd