

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ОТЧЕТ**  
**по лабораторной работе**  
**по дисциплине «Системное программирование в Linux»**  
**на тему «Демон для регулярного резервного копирования данных»**

Студент гр. 22.Б15-пу

\_\_\_\_\_

Осовский Н.С

Преподаватель

\_\_\_\_\_

Киямов Ж.У.

**Санкт-Петербург**  
**2023 г.**

## **Оглавление**

<b>1. Цель работы</b>	<b>3</b>
<b>2. Задача</b>	<b>3</b>
<b>3. Описание алгоритма</b>	<b>3</b>
<b>4. Описание программы</b>	<b>3</b>
<b>4.1. Описание функций</b>	<b>4</b>
<b>5. Рекомендации пользователя</b>	<b>4</b>
<b>6. Рекомендации программиста</b>	<b>5</b>
<b>7. Вывод</b>	<b>5</b>
<b>Приложение</b>	<b>6</b>

## **1. Цель работы**

Создание демона для автоматического регулярного резервного копирования данных с одного каталога в другой.

## **2. Задача**

Разработать конфигурационный файл, написать код демона на языке C++, интегрировать его в систему, создать механизм управления и мониторинга, создать документацию для пользователя, протестировать и оптимизировать демона, обеспечить безопасность.

## **3. Описание алгоритма**

1. Считывание конфигурации из конфигурационного файла.
2. В фоновом режиме ожидание наступления времени для следующей резервной копии.
3. Создание резервных копий файлов из исходного каталога в каталог для резервных копий с добавлением временной метки.
4. Журналирование выполнения операций в системный журнал.

## **4. Описание программы**

Данная программа предназначена для создания резервных копий файлов в указанной директории. Она состоит из нескольких файлов: `backup.cpp`, `backup.sh`, `backup_program`, `setup.sh` и `backup.conf`.

Файл `backup.cpp` содержит функцию `backupFile`, которая создает резервную копию одного файла. Функция принимает две строки - путь к исходному файлу и путь для создания резервной копии. Она открывает исходный файл и создает новый файл для резервной копии. Если оба файла открылись успешно, то копирует содержимое исходного файла в новый файл и выводит сообщение о создании резервной копии в стандартный вывод. Если же открыть файлы не удалось, то выводит сообщение об ошибке в стандартный поток ошибок.

Файл `backup_program` является основным исполняемым файлом программы. Он вызывает функцию `backupFiles` и передает ей пути к исходной директории и папке для создания резервных копий.

Файл `backup.conf` содержит настройки программы, такие как путь к исходной директории и папке для создания резервных копий.

Файл `backup.sh` содержит функцию `backupFiles`, которая создает резервную копию всех файлов в указанной директории. Функция принимает

две строки - путь к исходной директории и путь для создания директории с резервными копиями. Она проходит по всем файлам в указанной директории и, если это директория, создает соответствующую директорию в папке с резервными копиями и вызывает рекурсивно саму себя для этой директории. Если же это файл, то вызывает функцию backupFile для создания резервной копии этого файла в папке с резервными копиями. В процессе выполнения функции выводит сообщения о создании директорий и резервных копий в стандартный вывод.

Файл setup.sh является скриптом для управления программой: старт, перезапуск, отключение демона, просмотр логов, изменение конфигурационного файла

#### 4.1. Описание функций

В программе используются 2 функции

Таблица 5.1. Описание функций

Имя функции	Описание функции
backupFile	Функция принимает две строки - путь к исходному файлу и путь для создания резервной копии. Она открывает исходный файл и создает новый файл для резервной копии. Если оба файла открылись успешно, то копирует содержимое исходного файла в новый файл и выводит сообщение о создании резервной копии в стандартный вывод. Если же открыть файлы не удалось, то выводит сообщение об ошибке в стандартный поток ошибок.
backupFiles	Функция принимает две строки - путь к исходной директории и путь для создания директории с резервными копиями. Она проходит по всем файлам в указанной директории и, если это директория, создает соответствующую директорию в папке с резервными копиями и вызывает рекурсивно саму себя для этой директории. Если же это файл, то вызывает функцию backupFile для создания резервной копии этого файла в папке с резервными копиями. В процессе выполнения функции выводит сообщения о создании директорий и резервных копий в стандартный вывод.

#### 5. Рекомендации пользователя

Перед использованием демона необходимо настроить конфигурационный файл согласно своим потребностям.

Для управления и мониторинга демона необходимо использовать shell script.

При возникновении ошибок необходимо обратиться к системному журналу для получения информации об ошибке.

1. Создайте файл /etc/systemd/system/mybackup.servie

2. Вставьте в этот файл

[Unit]

Description=My script

After=network.target

[Service]

Type=simple

ExecStart=/путь\_к\_вашему\_скрипту/backup.sh

[Install]

WantedBy=multi-user.target

3. Запустите команды

sudo systemctl daemon-reload

4. Запустите демона с помощью setup.sh start

5. Просматривайте логи с помощью setup.sh log

## **6. Рекомендации программиста**

Для запуска программы необходим C++ 17. Запускать только на Linux, необходим установленный пакет systemd.

Библиотеки: iostream, fstream, filesystem.

## **7. Вывод**

Разработка демона для автоматического регулярного резервного копирования данных с одного каталога в другой является сложной задачей, которая требует знаний системного программирования в Linux. Однако, правильно написанный демон может значительно облегчить жизнь пользователям и обеспечить безопасность и сохранность данных.

## **Приложение**

<https://github.com/naelxd/linuxspbu>