**Лабораторная работа №9**

**Сервисы**

**Цель работы:** Получение практических навыков в работе с Windows-сервисами и Linux-демонами.

Перед выполнением задания прочитайте его полностью. Схематично представьте себе, что от вас требуется и только после этого выполняйте.

**Темы для предварительного изучения:**

* Функции для работы с сервисами в WinAPI ([ссылка](https://learn.microsoft.com/en-us/windows/win32/services/service-functions));
* Справочник по systemd ([ссылка](https://wiki.debian.org/systemd/Services));
* Пример скрипта для initd ([ссылка](https://gist.github.com/naholyr/4275302)).

**Постановка задачи для Windows:**

**Приложение Service09:**

Функционально сервисдолжен бытьаналогичен приложению **Lab-06с** в котором происходило отслеживание изменений в каталогах. (Если при сдаче лабы №6 в данном задании у вас отслеживались не все возможные события, то добавьте).

**Однако требуется обеспечить следующее поведение сервису**:

* При запуске сервиса – создаётся файл логирования с названием формата «<datetime>-dir.log»;
* Также должен создаваться файл с названием формата «<datetime>-srv.log» в который будет происходить логирование состояний самого сервиса.
* Файлы должны создаваться в каталоге, указанном в аргументах сервиса при его запуске (если каталог не существует, то он создаётся);
* Через аргументы сервиса также должен передаваться путь к каталогу события в котором будут отслеживаться;
* Если указанный для отслеживания каталог не существует, то сервис не запускается с логированием соответствующего сообщения;
* Требуется реализовать обработку возможных ошибок (при открытии файла и т.д.). Если ошибка не позволяет продолжить выполнение сервиса, то он должен быть переведён в состояние «Остановлен».

**Сервис должен поддерживать принятие и обработку следующих сигналов**:

* Запуск (поддерживается по умолчанию всеми сервисами) – при запуске сервис инициализирует все необходимые ресурсы (файлы и т.д.) и начинает свою работу по отслеживанию событий;
* Остановка (SERVICE\_ACCEPT\_STOP) – при остановке сервиса происходит остановка отслеживания событий и очистка всех ресурсов;
* Пауза (SERVICE\_ACCEPT\_PAUSE\_CONTINUE) – сервис просто приостанавливает работу по отслеживанию событий без очистки ресурсов;
* Возобновление (SERVICE\_ACCEPT\_PAUSE\_CONTINUE) – возобновляет работу по отслеживанию событий.

**Сервис должен записывать следующую информацию в \*-srv.log файл**:

* При создании каталога для файлов логирования, указанного в параметрах – Успех! <Имя сервиса> создал каталог <Путь к каталогу>;
* При обнаружении что каталог для файлов логирования указанный в параметрах существует – Успех! <Имя сервиса> обнаружил каталог <Путь к каталогу>;
* При обнаружении что каталог для отслеживания не существует – Попытка запуска провалена! <Имя сервиса> не смог обнаружить каталог <Путь к каталогу>;
* При успешном запуске – Успех! Сервис <Имя сервиса> запущен с параметрами <Параметры>;
* При смене состояния – Успех! Сервис <Имя сервиса> сменил состояние с <Текущее состояние> на <Новое состояние>
* При ошибках открытия файлов и т.д. – Попытка выполнения операции сервисом провалена! <Текст ошибки>.

**Сервис должен записывать в \*-dir.log файл** **информацию об объекте, с которым что-то произошло и что именно с ним произошло.**

Перед любым сообщением (и dir.log, и srv.log) должна присутствовать метка о времени, когда было записано сообщение (без даты).

**Приложение** **ServiceCtl:**

Приложение предназначено для управления сервисами. Семантика вызова данного приложения следующая:

**ServiceCtl <Имя сервиса> <Операция> [Дополнительные параметры]**

Приложение должно обеспечивать следующие операции (в скобках указаны названия и дополнительные параметры для данных операций):

* Создание сервиса – (Create, [путь к исполняемому файлу сервиса]);
* Запуск сервиса – (Start, [путь к каталогу за которым следим, путь к каталогу, где будут расположены файлы логов]);
* Остановка сервиса – (Stop);
* Удаление сервиса – (Delete);
* Приостановка – (Pause);
* Возобновление – (Continue);
* Вывод информации о сервисе – (Info).

Продемонстрировать управление службой через данное приложение, а также через стандартную утилиту **SC**.

**Постановка задачи для Linux:**

**Приложение Service09d:**

**Перед выполнением этой части работы проверьте какой менеджер служб у вас сейчас активен! (Если утилита systemctl выдаёт ошибку значит у вас Init, иначе Systemd)**

**В соответствии с этим требуется разработать скрипт файл для демона! Правила управления через отправку сигналов должны быть в нём!**

Функционально демондолжен бытьаналогичен приложению **Lab-06f** в котором происходило отслеживание изменений в каталогах. (Если при сдаче лабы №6 в данном задании у вас отслеживались не все возможные события, то добавьте).

**Однако требуется обеспечить следующее поведение сервису**:

* При запуске сервиса – создаётся файл логирования с названием формата «<datetime>-dir.log»;
* Также должен создаваться файл с названием формата «<datetime>-srv.log» в который будет происходить логирование состояний самого сервиса;
* Файлы должны создаваться в каталоге, указанном в конфигурационном файле демона (если каталог не существует, то он создаётся), если каталог не указан то, берётся значение по умолчанию «/var/log/service09d»;
* В конфигурационном файле демона также должен указываться путь к каталогу события в котором будут отслеживаться;
* Если указанный для отслеживания каталог не существует, то демон не запускается с логированием соответствующего сообщения;
* Требуется реализовать обработку возможных ошибок (при открытии файла и т.д.). Если ошибка не позволяет продолжить выполнение демона, то он должен быть завершён.

Конфигурационный файл расположить в каталоге, описанном в лекции.

**Демон должен поддерживать принятие и обработку следующих сигналов (путём регистрации обработчиков сигналов)**:

* Запуск (не совсем сигнал, тут описано поведение программы при запуске) – при запуске демон инициализирует все необходимые ресурсы (файлы и т.д.) и начинает свою работу по отслеживанию событий;
* Остановка (SIGTERM) – при остановке демона происходит остановка отслеживания событий и очистка всех ресурсов;
* Перезагрузка (SIGHUP) – демон повторно считывает файл конфигурации и выполняется согласно новым параметрам.

**Демон должен записывать следующую информацию в \*-srv.log файл**:

* При создании каталога для файлов логирования, указанного в параметрах – Успех! Демон <Имя демона> создал каталог <Путь к каталогу>;
* При обнаружении что каталог для файлов логирования указанный в параметрах существует – Успех! Демон <Имя демона> обнаружил каталог <Путь к каталогу>;
* При обнаружении что каталог для отслеживания не существует – Попытка запуска провалена! Демон <Имя демона> не смог обнаружить каталог <Путь к каталогу>;
* При успешном запуске – Успех! Демон <Имя демона> запущен с параметрами <Параметры>;
* При перезагрузке – Демон <Имя демона> теперь работает с параметрами <Параметры>.
* При завершении – Демон <Имя демона> завершил свою работу.
* При ошибках открытия файлов и т.д. – Попытка выполнения операции сервисом провалена! <Текст ошибки>.

**Сервис должен записывать в \*-dir.log файл** **информацию об объекте, с которым что-то произошло и что именно с ним произошло.**

Перед любым сообщением (и dir.log, и srv.log) должна присутствовать метка о времени, когда было записано сообщение (без даты).

Продемонстрировать работу демона, а также управление им через утилиту systemctl (для systemd) или service (для init).

Настроить автозапуск демона вместе с операционной системой.

**Требования к выполненной работе:**

* Разработанные приложения должны быть скомпилированы и протестированы как на Windows, так и на ОС из семейства Linux.
* Для генерации и сборки проекта использовать средство CMake.
* Для компиляции файлов при сборке проекта использовать компилятор Clang.

**Вопросы для контроля:**

1. Что такое служба?
2. Какие бывают службы?
3. Чем можно охарактеризовать службы?
4. Что такое SCM?
5. Где располагается информация о сервисе?
6. Что такое Linux-демон?
7. Что такое Init и Systemd?