

2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

В структуре разрабатываемого дипломного проекта можно выделить следующие блоки (см. схему ГУИР.400201.170 С1):

- блок интерфейса пользователя;
- блок описания команд;
- блок настройки приложения;
- блок журнала событий;
- локальная база данных;
- блок управления правилами;
- блок мониторинга файловой системы;
- блок пересылки сообщений;
- блок анализа файлов;
- блок взаимодействия с монитором;

Блок интерфейса пользователя предназначен для предоставления пользователю возможности конфигурировать работу приложения, настраивать правила сортировки и перемещения для пользовательских директорий и просматривать отчёты о результатах работы приложения. Так же блок интерфейса пользователя позволяет своевременно выводить сообщения об ошибках, произошедших во время работы приложения. Результатом работы пользователя с данным блоком является команда в определённом формате, которая передаётся блоку описания команд, для анализа и выполнения.

Блок описания команд предназначен для приёма команд от интерфейса пользователя и их интерпретации. Он необходим для того что бы работа приложения не зависела от вида интерфейса пользователя. Блок описания команд предоставляет следующие команды:

- создание, чтение, обновление и удаление настроек приложения;
- создание, чтение, обновление и удаление правил управления файлами в пользовательских директориях;
- просмотр записей в журнале событий;
- немедленное выполнение заданного действия для заданной директории;
- откат состояния отслеживаемых директорий до заданной записи в журнале событий;
- импорт и экспорт настроек приложения;
- импорт и экспорт правил по управлению файлами;
- импорт и экспорт настроек и правил вместе;
- запуск и отключение мониторинга;

Блок описания команд взаимодействует с блоком управления правилами, блоком настройки приложения и блоком журнала событий. К блоку журнала событий имеется только доступ на чтение.

Блок настройки приложения управляет действиями, которые не

относятся к основной функциональности приложения, а влияют на его работу в конкретной операционной системе. К таким действиям относится настройка запуска отдельных блоков приложения как демонов или сервисов в операционной системе, проверка наличия всех необходимых библиотек, установка и удаление необходимых переменных окружения, проверка обновлений и т.д. Так же с помощью этого блока пользователь имеет возможность настроить специальное поведение приложения, которое будет учитывать низкую скорость жёсткого диска компьютера или высокую частоту файловых операций. Блок настройки приложения сохраняет все параметры, необходимые для работы, в файле, в специальном формате, и имеет возможность экспортировать или импортировать этот файл. Данный блок принимает изменения со стороны блока описания команд и используется блоком взаимодействия с монитором.

Локальная база данных представляет собой реляционную базу данных с высокими требованиями к скорости записи. Скорость чтения значительной роли не играет, так как в блоках, которые используют базу данных и для которых скорость чтения является критичным параметром реализован кэш.

Блок журнала событий предназначен для конвертации событий, созданных приложением, в записи журнала и сохранения этих записей в локальную базу данных. Данный блок позволяет пользователю получить информацию о том, что приложение сделало с файлами, которые подпали под правила, заданные пользователем. Так же, поскольку записи событий сохраняются последовательно, есть возможность отменить некоторые (не все) действия, сделанные приложением после определённой записи. Это даёт приложению дополнительную надёжность, так как в случае сбоя или непредвиденного поведения пользователь получает возможность вмешаться в то, что сделало приложение и исправить ошибку. События, которые служат основой для записей журнала генерируются блоком взаимодействия с монитором файловой системы.

Блок управления правилами предоставляет глобальный интерфейс по работе с правилами обработки файлов. Основная его функциональность - вычисление правила или группы правил, которые применимы к файлу, который сгенерировал событие в файловой системе. Основные ситуации, которые должны обрабатываться данным блоком:

- 1) Корректная обработка приоритетов правил. Правила с большей специфичностью должны применяться в первую очередь, а в некоторых ситуациях полностью перекрывать правила, для которых свойства подпадающих под них файлов описаны в более общем виде.
- 2) Корректная обработка пересечений правил. Если к файлу применимы два правила, которые предусматривают одинаковое действие – это действие должно примениться только один раз.
- 3) Оптимизация правил. Если за правилами, которые предусматривают какие-либо действия над файлом, идёт правило, которое должно этот файл безвозвратно удалить – правила перед удалением удаляются.

- 4) При удалении отслеживаемой директории, удаляются все правила, связанные с этой директорией.
- 5) При изменении имени отслеживаемой директории изменяются все правила, связанные с этой директорией.
- 6) При появлении правил, которые добавляют новую директорию для отслеживания или при удалении всех правил, связанных с одной из существующих директорий, данный блок должен оповестить монитор файловой системы, используя для этого интерфейс блока взаимодействия с монитором.

Также этот блок полностью ответственен за хранение правил и знает, как и когда их нужно добавлять, обновлять или удалять и содержит в себе уровень кэша над локальной базой данных, в который правила считываются на старте приложения. Уровень кэша необходим для того что бы обеспечить необходимую скорость обработки событий файловой системы. Потеря данных о заданных правилах не допускается, поэтому кэш сквозной - запись правила в кэш приводит к немедленной записи в базу данных.

Блок управления правилами используется блоком взаимодействия с монитором файловой системы и блоком описания команд. В качестве входной информации, по которой нужно определить правила, применимые к файлу, используются данные о событии в файловой системе и данные от блока анализа файлов. Новые правила так же добавляются в специальном формате.

Блок мониторинга файловой системы представляет собой программу в двух версиях – в виде демона Linux или сервиса Windows. Эта программа производит мониторинг файловой системы для отслеживаемых директорий, и когда в файловой системе происходит какое-то событие она конвертирует его в сообщение текстового формата и отправляет его через блок пересылки сообщений блоку взаимодействия с монитором. Данным блоком отслеживаются следующие события:

- появление нового файла в отслеживаемой директории;
- изменение имени файла в отслеживаемой директории;
- изменение имени отслеживаемой директории (приводит к обновлению правил для этой директории);
- удаление файла из отслеживаемой директории;
- изменение файла в отслеживаемой директории;
- файл был перемещён из отслеживаемой директории;
- файл был помещён в отслеживаемую директорию;
- отслеживаемая директория была удалена (приведёт к отмене всех правил, связанных с данной директорией);
- отслеживаемая директория была перемещена (приведёт к изменению всех правил, связанных с данной директорией)

Так же данный блок способен принимать сообщения о появлении новых директорий для отслеживания от основного приложения (конкретно – блок взаимодействия с монитором). Для мониторинга событий используется

библиотека inotify для ОС Linux, и функции WIN64 API для ОС Windows.

Так как блок мониторинга файловой системы работает как отдельная программа, ей нужно средство связи с основным приложением. В качестве этого средства связи выступает блок пересылки сообщений. Блок пересылки сообщений представляет собой библиотеку, предоставляющую возможность обмена сообщениями в рамках распределённого приложения. В данном дипломном проекте для этой цели будет использована библиотека ZeroMQ. Для обмена сообщениями эта библиотека использует сокет без гарантированной доставки сообщений.

Блок анализа файлов предназначен для получения всех метаданных файла, сгенерировавшего событие в отслеживаемой директории. Правила, задаваемые пользователем, могут быть привязаны к свойствам этих метаданных. К интересующим приложение свойствам файлов относятся:

- 1) Имя, расширение, дата создания (базовый анализ).
- 2) Исполнитель, альбом, жанр (анализ музыкальных файлов).
- 3) Тематика, язык, жанр (анализ электронных книг).

Результат анализа файла вместе с данными о событии служат основной информацией для определения правил, которые в свою очередь определяют набор необходимых действий над данным файлом. Блок анализа файлов используется только блоком взаимодействия с монитором файловой системы.

Блок взаимодействия с монитором файловой системы является центральным блоком приложения. Он предназначен для передачи команд блоку мониторинга файловой системы и обработки событий в отслеживаемых директориях, пришедших от этого блока.

Команда конвертируется в текстовый формат и передаётся через блок передачи сообщений. Передаются следующие команды:

- включение монитора;
- выключение монитора;
- добавление новой директории для отслеживания с настройками отслеживания;
- удаление директории из списка отслеживаемых;

События приходят от блока мониторинга также в текстовом формате. Порядок обработки события следующий:

- 1) Блок получает сообщение от блока мониторинга файловой системы через блок пересылки сообщений.
- 2) Блок создаёт отдельный поток обработки сообщения, и передаёт ему сообщение на обработку.
- 3) Сообщение переводится из текстового формата в объектный формат приложения.
- 4) По типу события определяется может ли приложение обработать его прямо сейчас и нужен ли анализ файла, сгенерировавшего событие.
- 5) Если анализ файла нужен он проводится с помощью блока анализа файлов.

- 6) Далее данные о событии и, опционально, результаты анализа файлов отправляются блоку управления правилами. В ответ приходит список правил, применимый к данному файлу.
- 7) Блок по порядку выполняет действия, предусмотренные правилами.
- 8) Данные о событии и действиях записываются в журнал событий.
- 9) Поток обработки сообщения освобождает ресурсы.

Также, поскольку этот блок является центральным и выполняет все действия над файлами, для него очень важна правильная обработка ошибок, поскольку всего одна необработанная ошибка может остановить работу всего приложения.

В этом разделе были подробно рассмотрены все основные логические блоки приложения и связи между ними, которые явно описывают то как пользователь будет взаимодействовать с программой и как компоненты программы будут взаимодействовать между собой. Система спроектирована так, что на данном этапе не зависит от целевой операционной системы и языка программирования, который может быть использован для реализации данного приложения.