4 ОПИСАНИЕ АЛГОРИТМА ПРОГРАММЫ

Для выполнения требований технического задания разработан алгоритм передачи данных из открытой сети в закрытую. Алгоритм работы программы состоит из двух блоков: приемного и передающего. Блок схемы алгоритмов приведены в приложениях Б, В, Г, Д.

Перед началом работы вне зависимости от функционального назначения программы, происходит вызов программы инициализации, представленный в приложении В.

Программа пытается найти файл конфигурации, и в случае, если она его не находит, создает новый файл конфигурации в каталоге после чего заполняет его стандартной конфигурацией.

Для настройки программы можно использовать аргументы командной строки или изменять файл конфигурации. В случае использования аргументов командной строки, программа самостоятельно обновит файл конфигурации.

Общие настройки для передающего и принимающего режимов работы программы:

– Выбор алгоритма вычисления контрольной суммы;

– Настройка местонахождения рабочих каталогов.

Для передающего режима доступны:

– Настройка адреса и порта для отправки;

– Возможность подписаться на системные события в рабочем каталоге, для мгновенной обработки новых изменений;

– Установка максимального размера пакета при передаче;

– Настройка структуры токена идемпотентности;

– Настройка имени передатчика для опознания на приемной стороне;

– Настройка частоты сканирований каталога на изменения.

Для принимающего режима доступны:

– Выбор операции при удалении и модификации файла;

– Настройка времени хранения удаленных и измененных файлов в каталоге «Old»;

– Настройка имени источника;

– Установка времени хранения токенов идемпотентности;

– Возможность вызова произвольной команды при наступлении ошибки для уведомления администратора.

После загрузки конфигурации происходит поиск файла с текущим состоянием каталога. В случае если он не был обнаружен, создается новый файл состояния каталога.

По завершению настройки, программа приступает к работе в основном режиме.

Рассмотрим работу передающей части программы. Алгоритм ее работы приведен в приложении Г. После инициализации происходит ожидание следующего периода синхронизации, после чего, каталог сканируется на предмет изменений, путем сравнения с прошлым его состоянием.

Для этого, на каждый файл, находящийся в каталоге, создается запись с его именем, временем последнего изменения, и прочими метаданными. В случае изменения файла, он добавляется в очередь операций. По завершению сканирования начинается процесс обработки очереди операций.

При обработке каждого изменения, перед началом отправки происходит ожидание завершения работы с файлом, для предотвращения ситуации отправки неполного документа.

По завершению модификации файла пользователем, происходит его блокировка на запись, для предотвращения изменения файла в процессе отправки.

Далее происходит вызов подпрограммы отправки изменения, алгоритм которой приведен в приложении Д.

Перед отправкой происходит сбор метаданных файла, подсчет его контрольной суммы, сохранение времени начала отправки файла, после чего файл разделяется на части, и они отправляются в очередь отправки.

Когда программа начинает обрабатывать очередь отправки, происходит подсчет контрольной суммы части файла, генерируется токен идемпотентности, происходит упаковка файла в структуру для отправки данных.

В данной структуре содержится информация:

– Имя источника;

– Имя файла;

– Тип изменения;

– Время изменения файла;

– Номер блока;

– Контрольная сумма файла;

– Контрольная сумма блока;

– Токен-идемпотентности;

– Длинна данных;

– Массив данных.

После упаковки происходит отправка пакета данных по сети, после чего, алгоритм проделывает тоже самое с каждым блоком файла.

По завершению отправки файла в сеть, происходит запись в журнал о завершении отправки файла, обновляется состояние каталога, снимается ограничение на модификацию с отправляемого документа.

Рассмотрим алгоритм принимающей подпрограммы, после инициализации, программа ожидает получение пакета на заданном порте. Когда пакет приходит, происходит проверка на соответствие источника пакета, заданному источнику в файле конфигурации. Если имена не совпадают, пакет сбрасывается.

Если имена совпали, происходит проверка на наличие токена идемпотентности в кэше. Если токен присутствует, то значит, пакет уже обрабатывался, и его нужно сбросить.

В случае отсутствия токена в кэше, происходит его добавление, проверка кэша на наличие устаревших записей и последующая очистка.

После этого, происходит проверка, соответствует ли текущее состояние каталога, целевому. Если без обработки пакета, каталог будет соответствовать целевому, то пакет будет сброшен.

Далее проверяется контрольная сумма массива данных, если контрольная сумма не верна, пакет сбрасывается.

После этого, происходит обработка пакета в зависимости от типа изменения.

В случае если пакет имеет тип «Создать» или «Модификация» происходит проверка на то, существует ли такой файл в рабочем каталоге. Если существует происходит его перенос в каталог «Old» или удаление в зависимости от конфигурации.

Далее проверяется первый ли это пакет для файла, если да, происходит создание нового файла в каталоге «Downloading» где создается пустой файл, который записывается нулями до целевого размера.

После проверок происходит замена нулей в файле на данные из пришедшего пакета.

По завершению замены, проверяется, все ли части файла пришли, если да, происходит проверка контрольной суммы и в случае успеха, файл переносится в рабочий каталог, где с ним могут взаимодействовать пользователи приватной сети.

Текущее состояние каталога обновляется в кэше, после чего, происходит очистка каталогов «Old» и «Downloading» от устаревших записей.

После этого, алгоритм принимающей стороны возвращается к ожиданию новых пакетов.

По-особому в данной цепочке обрабатывается тип пакета «Удалить». После всех проверок, происходит перенос файла в каталог «Old» или удаление в зависимости от конфигурации.

В случае логической ошибки, например, попытка удаления отсутствующего файла или модификация еще не существующего файла, можно настроить желаемое поведение, а также уведомить администратора о случившемся.

Листинг программы приведен в приложении Ж. Программа написана на языке С++, асинхронные операции ввода вывода реализованы на базе библиотеки Boost ASIO. Использование Boost позволило написать кроссплатформенный код, который может быть скомпилирован для любой платформы.