**ВВЕДЕНИЕ**

Иногда приходится переместить некоторую информацию из одного места в другое. Однако некоторые файлы занимают слишком много места, и их пересылка через ограниченное по скорости сетевое подключение займет слишком много времени. Или место назначения, куда нужно скопировать файл, ограниченно в объеме. В таких случаях просто необходимо как-то сократить объем файла. Для этой задачи пользователи ПК используют архиваторы файлов и директорий. Все архиваторы делятся на те, которые применяют алгоритм сжатия с потерями, и те, что сжимают без потерь. Хоть на первый взгляд и кажется, что сжатие с потерями не имеет смысла, ведь, сжав, например, текстовый файл с каким-то текстом, разархивировав его мы не сможем получить осмысленный текст. Однако эти программы активно применяются в области сжатия аудиозаписей, где некоторая потеря информации не сильно сказывается на восстановленном файле.

Степень сжатия данных у алгоритма без потерь ниже, чем у сжатия с потерями. Однако в курсовом проекте используется именно алгоритм сжатия без потерь по алгоритму Хаффмана, что позволит сжимать все виды файлов и вновь использовать их после восстановления. По этому алгоритму каждый символ в файле заменяется его битовой кодировкой, которая создается с помощью построения сбалансированного бинарного дерева, где в листьях хранятся символы и количество их повторений в файле, а в узлах хранятся числа, которые равны сумме чисел в дочерних узлах.

В конечном итоге программа позволяет сжимать файлы и директории, создавая архив. Также программа способна провести разархивацию сжатой информации, восстановив исходные данные без потерь. Для того, чтобы восстановить данные, необходимо как-то восстановить битовые коды символов. Поэтому в начале архива помещается информация для программы, в которую записывается информация о необходимых папках, а перед каждой сжатой записью файла находится словарь кодировок, где к каждому встреченному в исходном файле байту сопоставляется новый битовый код. Самая высшая степень сжатия по данному алгоритму у текстовых файлов. Это обусловлено тем, что в текстовых файлах вы редко встретите более 130 различных видов байт, что позволяет построить относительно небольшое дерево. С помощью дерева каждый байт, состоящий из 8 бит, может быть заменен другим битовым кодом. К примеру, самым часто встречающимся символом в текстовых файлах является пробел. Однако он занимает столько же места, сколько и любой другой символ. При построении дерева учитывается число повторений символа в файле, таким образом символ пробела окажется в самой вершине дерева. Особенность бинарного дерева в том, что путь к каждому листу дерева уникален. Если взять проход влево за единицу, а проход вправо за ноль, то путь от вершины к каждому узлу сгенерирует свой уникальных бинарный код. Поэтому путь к редко встречающимся символам будет несколько длиннее восьми бит, обычно около 12, однако в это же время путь к пробелу может состоять всего из двух бит. А учитывая то, что пробелы встречаются намного чаще, чем те символы, код которых становится больше восьми бит, получается сжать информацию.

В общем случае архиватор хорошо показывает себя и с другими типами информации. Так некоторые изображения получается сжать в два раза. Но основной проблемой является то, что аудиозаписи, видеофайлы и многие другие форматы информации содержат в себе все 256 различных возможных байт, из-за этого не всегда получается уменьшить размер файла.

Работа программы сосредоточена в трех потоках: потоке архивации, потоке разархивации и основном потоке приложения, где обрабатывается графический интерфейс, а также ввод пользователя и вывод информации для него.

Программ, способных на архивацию данных достаточно много, но в большинстве своем это довольно старое ПО с малоудобным интерфейсом. Поэтому основной целью данного курсового проекта является создание не только быстрого, легкого, но и удобного для пользователя приложения по архивации файлов и директорий без потерь информации.

Для разработки данного курсового проекта был выбран известный фреймворк Qt. Сам по себе он очень удобен для разработчика, так как содержит в себя обширную библиотеку классов. Также Qt нацелен на создание пользовательского интерфейса. Языком написания проекта является C++, он хорошо знаком с предыдущего года обучения, а также довольно быстр, по сравнению с другими языками программирования, что позволит ускорить работу архивации и разархивации данных.