Алгоритм работы программы:

1. Производится анализ числа аргументов командной строки:

Если их нет, то сообщить об их отсутствии и вернуть 1,  
иначе продолжаем анализ;

1. Проверяем первый аргумент командной строки:  
   Если он равен "compress", то выполняем сжатие,

если он равен "decompress", то выполняем распаковку,

иначе сообщаем об ошибке и возвращаем 1;

Сжатие:

1. Проверяем число аргументов:  
   Если из не менее 3, то выполняем сжатие,  
   Иначе сообщаем об ошибке и возвращаем 1;
2. Открываем файл, который будем сжимать.
3. Создаем выходной файл, в котором будет результат сжатия.
4. Открываем поток сжатия GZipStream с указанием выходного файла и типом режима: Сжатие.
5. Выводим сообщение о начале сжатия.
6. Инициализируем 2 потока обработки входного файла и передаем им **функцию сжатия**.
7. Вывод сообщения о успешном завершении операции.
8. Освобождение ресурсов.
9. Возвращение 0 основной функцией, как знак успешного выполнения операции сжатия.

Функция сжатия:

1. Каждый из потоков синхронизирован по блокировке взаимного исключения (оператор lock).
2. Производится анализ размера входного файла:
3. Если его размер меньше 70% свободного ОЗУ, то размер буфера равен размеру файла, иначе каждому потоку относительно текущего состояния свободного места ОЗУ выделяется размер буфера данных. (5% от свободного ОЗУ).
4. Функция, которая задана потокам, выполняет считывание из входного файла данные, затем сжимает и результат сжатия записывает в конец выходного файла.
5. Считывание производиться до тех пор, пока не дойдем до конца файла.

Распаковка:

1. Проверяем число аргументов:  
   Если из не менее 3, то выполняем распаковку,  
   Иначе сообщаем об ошибке и возвращаем 1;
2. Открываем файл, который будем распаковывать.
3. Создаем выходной файл, в котором будет результат распаковки.
4. Открываем поток сжатия GZipStream с указанием выходного файла и типом режима: Распаковка.
5. Выводим сообщение о начале распаковки.
6. Инициализируем 2 потока обработки входного файла и передаем им **функцию распаковки**.
7. Вывод сообщения о успешном завершении операции.
8. Освобождение ресурсов.
9. Возвращение 0 основной функцией, как знак успешного выполнения операции распаковки.

Функция распаковки:

1. Каждый из потоков синхронизирован по блокировке взаимного исключения (оператор lock).
2. Производится анализ размера входного файла:
3. Если его размер меньше 70% свободного ОЗУ, то размер буфера равен размеру файла, иначе каждому потоку относительно текущего состояния свободного места ОЗУ выделяется размер буфера данных. (5% от свободного ОЗУ).
4. Функция, которая задана потокам, выполняет считывание из входного файла данные, затем распугивает данные и результат записывает в конец выходного файла.
5. Считывание производиться до тех пор, пока не дойдем до конца файла.