Когда размер сортируемых данных невелик, их можно хранить в оперативной памяти, которая обладает быстрой скоростью доступа и обеспечивает быструю обработку данных. В таком случае можно использовать алгоритмы сортировки, которые выполняются в памяти, такие как сортировка вставками, сортировка пузырьком или быстрая сортировка.

Если размер сортируемых данных большой и не помещается в оперативную память, то для сортировки используются специальные методы, которые основаны на работе с файлами. В таком случае данные представляются в виде файла, который может быть разделен на части для ускорения сортировки.

Доступ к элементам файла производится в определенной последовательности, так как файлы хранятся на диске, где доступ к данным происходит последовательно.

Сортировка больших файлов может выполняться с помощью алгоритма сортировки слиянием. В таком случае файл разбивается на несколько частей, которые сортируются независимо друг от друга. Затем эти отсортированные части объединяются в единый файл с помощью операции слияния. Этот процесс повторяется до тех пор, пока не будет получен полностью отсортированный файл. Сортировка слиянием может быть эффективной для сортировки больших файлов, но требует дополнительной памяти для хранения промежуточных результатов.

Слияние (merge) - это алгоритм сортировки, который объединяет две или более отсортированные последовательности в одну упорядоченную последовательность. Он осуществляет выбор элементов из каждой из последовательностей в порядке возрастания или убывания и помещает их в результирующую последовательность.

В простом случае, когда мы объединяем две отсортированные последовательности A и B, мы начинаем с двух указателей, один указывает на начало последовательности A, а другой на начало последовательности B. Затем мы сравниваем первые элементы из обеих последовательностей, выбираем наименьший (если сортируем по возрастанию) и добавляем его в результирующую последовательность. Указатель на выбранный элемент сдвигается в соответствующей последовательности на одну позицию вправо. Мы повторяем этот процесс до тех пор, пока одна из последовательностей не будет полностью использована, а все ее элементы не будут добавлены в результирующую последовательность. Затем мы просто добавляем оставшиеся элементы из другой последовательности в результирующую последовательность.

Работа с файлами в программировании включает в себя чтение, запись и обработку данных, хранящихся в файлах на компьютере.

Файловый тип - это способ представления данных в файле. Файл может содержать произвольную последовательность элементов, таких как числа, символы, строки и т.д. Длина файла может изменяться в процессе выполнения программы, в зависимости от количества элементов, которые были записаны или удалены из файла.

Текстовые файлы - это файлы, содержащие символьные данные, которые интерпретируются как текст. Текстовые файлы могут содержать различные форматирования, такие как перенос строки, табуляцию и пробелы. Текстовые файлы могут быть прочитаны и изменены с помощью текстовых редакторов.

Файлы с типом - это файлы, которые имеют определенный формат и предназначены для обработки определенным приложением. Файлы с типом могут содержать данные, такие как изображения, звуковые файлы и видео. Файлы с типом могут быть обработаны только теми приложениями, которые могут интерпретировать их формат.

Файлы без типа - это файлы, которые не имеют определенного формата и содержат произвольные данные, которые могут быть интерпретированы по-разному в зависимости от контекста. Файлы без типа могут содержать текстовые данные, бинарные данные или любую другую информацию, которая может быть хранится в файле. Такие файлы могут быть обработаны различными приложениями в зависимости от способа интерпретации данных.

Окно файла (или указатель файла) - это позиция в файле, с которой выполняются операции чтения и записи данных. Оно представляет собой указатель на текущую позицию в файле и определяет элемент файла, который доступен для чтения или записи.

При открытии файла, операционная система устанавливает указатель файла на начало файла, т.е. на первый элемент в файле. После этого, указатель может изменяться в процессе работы с файлом, в зависимости от выполняемых операций.

Например, если программа выполняет операцию чтения данных из файла, то указатель файла будет смещаться на количество прочитанных байтов, чтобы при следующей операции чтения, программа начинала считывать данные с того места, где закончилось предыдущее чтение.

Аналогично, при операции записи данных в файл, указатель файла указывает на позицию, куда нужно записать данные. Если данные уже были записаны в файл, то новые данные будут записаны после старых, и указатель файла будет указывать на позицию следующую за последней записанной позицией.

Использование окна файла упрощает работу с файлами, так как позволяет устанавливать точку начала чтения или записи данных, а также контролировать текущую позицию в файле, что позволяет эффективно управлять процессом чтения и записи данных.

В Delphi процедура AssignFile используется для связывания файла с переменной в программе. Эта процедура должна быть вызвана перед выполнением любых операций с файлом, таких как чтение или запись.

Файлы с типом - это файлы, которые содержат данные одного типа (например, только числа или только строки). При объявлении переменной типа File в Delphi, можно указать тип данных, которые будут содержаться в файле, например:

Здесь переменная File содержит данные типа Real. Это означает, что при записи или чтении из файла, в нем будут использоваться только вещественные значения.

Важно понимать, что при работе с файлами с типом необходимо следить за типами данных, которые записываются в файл и считываются из него. Если тип данных не совпадает с указанным при объявлении переменной типа File, то возможны ошибки или некорректные результаты.

Процедура Rewrite(F) в языке программирования Delphi используется для создания и открытия нового файла F для записи данных. Если файл уже существует, то он будет перезаписан. Если файл был ранее открыт, то он будет закрыт и затем повторно открыт в режиме записи.

После вызова процедуры Rewrite указатель в файле устанавливается на первую позицию, так что следующая операция записи будет произведена в начале файла. Если в файле уже содержатся данные, то они будут перезаписаны.

Процедура Write в языке программирования Delphi позволяет записывать данные в файл. Первым параметром процедуры является файловая переменная F, которая представляет собой открытый файл. Затем следуют параметры V1, V2,...,VN - значения, которые нужно записать в файл.

Процедура Reset(F) в Delphi предназначена для открытия существующего файла с именем, которое было предварительно присвоено переменной F процедурой Assign. Если файл с указанным именем не существует, то возникает ошибка.

При открытии файла с помощью процедуры Reset(F) указатель файла устанавливается на первую позицию, так что первый элемент файла становится доступным для чтения. Если файл был предварительно открыт, то он закрывается и повторно открывается.

Процедура Reset(F) может быть применена к файлу любое количество раз, но при этом содержимое файла не изменяется. Это означает, что если файл был изменен другими процедурами или приложениями после того, как был предварительно открыт в программе с помощью Assign, то при выполнении процедуры Reset(F) в программе будут использоваться измененные данные файла.

Процедура Read (F, V [, V2, …, VN]) является процедурой чтения данных из файла F в переменные V1, V2, …, VN. При вызове процедуры Read, указатель файла F перемещается на следующую позицию, после той, которая содержала последний считанный элемент.

Функция EoF (F) используется для проверки, достигнут ли конец файла F. Если функция возвращает true, это означает, что последняя операция чтения данных считала последний элемент файла. Функция EoF может быть использована в циклах чтения файла, чтобы определить момент, когда нужно остановиться.  
  
 Функция FilePos (F) возвращает номер текущей позиции указателя файла F. Это означает, что функция позволяет определить, на каком месте в файле находится указатель в данный момент. Если указатель находится в начале файла, то функция вернет значение 0. Например, если файл содержит 10 элементов, а указатель находится на 6-м элементе, то вызов функции FilePos(F) вернет значение 6.

Функция FileSize (F) возвращает текущий размер файла F в байтах, то есть количество байт, занимаемых файлом. Типом результата функции является целое число (integer). Если файл пустой, функция вернет значение 0. Например, если размер файла составляет 100 байт, то вызов функции FileSize(F) вернет значение 100.

Процедура CloseFile(F) в Delphi используется для закрытия открытого файла, где F - переменная типа file, которая была использована для открытия файла с помощью процедуры AssignFile. После того, как файл был обработан, необходимо закрыть его с помощью процедуры CloseFile, чтобы освободить ресурсы и избежать ошибок при повторном использовании этого файла. Если файл уже был закрыт с помощью процедуры CloseFile или не был открыт вообще, то повторный вызов процедуры CloseFile(F) для этого файла не будет иметь никакого эффекта и не вызовет ошибку.

Процедура Erase(F) в Delphi используется для попытки стирания (удаления) файла F, имя которого должно быть назначено с помощью процедуры AssignFile. Если файл не существует, то при вызове процедуры Erase(F) поднимается исключение

na - количество элементов в массиве a

nb, nc - количество элементов в массивах b и c, соответственно

p - размер блока, на который разбивается массив для слияния

temp - временная переменная для хранения считываемых значений из файла

posb, posc - указатели на текущий элемент в массивах b и c, соответственно

valb, valc - значения текущих элементов в массивах b и c, соответственно

pos - указатель на текущий элемент в массиве a

revc - счётчик элементов в текущем блоке

rev - флаг, указывающий на то, в каком массиве (b или c) будут записываться элементы на данной итерации

isActualB, isActualC - флаги, указывающие на то, является ли значение в valb или valc актуальным

Supp: TSupp - переключатель поддержки, позволяющий определить, какой из двух файлов (F1 или F2) будет использоваться в качестве текущего при разбиении файла на серии. Может принимать значения 1 или 2.

Series: Integer - счетчик серий. Используется для определения того, когда завершена сортировка файла.

Tmp1, Tmp2, Tmp: Integer - переменные, которые используются для хранения значений, считанных из файла F и временных значений при слиянии серий.

Save1, Save2: Integer - переменные, используемые для хранения последних записанных значений в каждом из файлов F1 и F2. Необходимы при слиянии серий.

F, F1, F2: TFile - файловые переменные, используемые для открытия и работы с файлами.