Есть такое понятие как нетипизированный файл. Это такой файл который содержит разнородные элементы. Например файл EXE - вначале он имеет заголовок, затем двоичный код, в конце какие-то ресурсы. Все части файла имеют разную длину и разную структуру. Тут уже обратится к произвольному элементу сложно, обычно надо вначале узнать где этот элемент находится, подчас это записано в предыдущем куске информации. Работа с такими файлами достаточно сложна и требует вручную разработки алгоритмов его чтения, но в связи гибкостью структуры и компактностью такие файлы составляют большинство. Для работы с нетипизированными файлами используют процедуры BlockRead и BlockWrite, которые позволяют читать/писать произвольное количество байт. Привожу пример пользования этими функциями из справки по Дельфи:

|  |
| --- |
| **var** |
| FromF, ToF: **file**; |
|  |
| NumRead, NumWritten: Integer; |
|  |
| Buf: **array**[1..2048] **of**Char; |
|  |
|  |
| **begin** |
| **if**OpenDialog1.Execute **then***{ Display Open dialog box }* |
|  |
|  |
| **begin** |
|  |
|  |
| AssignFile(FromF, OpenDialog1.FileName); |
|  |
| Reset(FromF, 1); *{ Record size = 1 }* |
|  |
| **if**SaveDialog1.Execute **then***{ Display Save dialog box}* |
|  |
|  |
| **begin** |
|  |
|  |
| AssignFile(ToF, SaveDialog1.FileName); *{ Open output file }* |
|  |
| Rewrite(ToF, 1); *{ Record size = 1 }* |
|  |
| Canvas.TextOut(10, 10, 'Copying ' + IntToStr(FileSize(FromF))+ ' bytes...'); |
|  |
| **repeat** |
|  |
|  |
| BlockRead(FromF, Buf, SizeOf(Buf), NumRead); |
|  |
| BlockWrite(ToF, Buf, NumRead, NumWritten); |
|  |
|  |
| **until**(NumRead = 0) **or**(NumWritten <> NumRead); |
|  |
| CloseFile(FromF); |
|  |
| CloseFile(ToF); |
|  |
|  |
| **end**; |
|  |
|  |
| **end**; |
|  |
|  |
| **end**; |