



Figure 19.1 Illustration of a disk partitioned into three areas for the Xinu file system.

The first sector of the disk holds a directory that contains a list of file names along with a pointer to the list of index blocks for the file. The directory also contains two other pointers: one to a list of free (unused) index blocks and another to a list of free data blocks. The directory entry for a file also contains an integer that gives the current size of the file measured in bytes.

Following the directory, the disk contains an index area that holds a set of *index blocks*, abbreviated *i-blocks*. Each file has its own index, which consists of a singly-linked list of index blocks. Initially, all index blocks are linked onto a free list from which the system allocates one as needed; index blocks are only returned to the free list if a file is truncated or deleted.

Following the index area, remaining blocks of the disk comprise a data area. Each block in the data area is referred to as a *data block*, abbreviated *d-block*, because a block contains data that has been stored in a file. Once a data block has been allocated to a file, the block only contains data. That is, a data block does not contain pointers to other data blocks, nor does it contain information that relates the block to the file of which it is a part; all such information resides in the file's index.

Similar to index blocks, when a disk is initialized, the data blocks are linked onto a free list. The file system allocates data blocks from the free list as needed, and returns data blocks to the free list when a file is truncated or deleted.

Figure 19.2 illustrates the conceptual data structure used for a Xinu file system. The figure is not drawn to scale: in practice a data block is much larger than an index block and occupies one physical disk block. The important idea is that the data structure illustrated in the figure resides on disk. We will see that at any given time, only a few pieces of the structure are present in memory — the file system must create and maintain an index without reading the structure into memory.

19.5 Implementation Of The Index Manager

Conceptually index blocks form a randomly accessible array that is mapped onto a contiguous area of the disk. Thus, index blocks are numbered from zero through K , and the software uses the number to refer to a given index block. Because an index block is smaller than a physical disk block, the system stores seven index blocks into each physical block, and the software handles the details of reading and writing an individual index block.