**FILE DESIGN**

A ﬁle system provides the machinery to support the project tasks. At the highest level a ﬁle system is a way to organize, store, retrieve, and manage information on a permanent storage medium such as a disk. File systems manage permanent storage and form an integral part of all operating systems. There are many different approaches to the task of managing permanent storage. At one end of the spectrum are simple ﬁle systems that impose enough restrictions to inconvenience users and make using the ﬁle system difficult. In deciding what type of ﬁling system is appropriate for a particular operating system, we must weigh the needs of the problem with the other constraints of the project. The two basic abstractions of files and directories form the basis of what a ﬁle system can operate on. There are many operations that a file system can perform on ﬁles and directories. All ﬁle systems must provide some basic level of support. Beyond the most basic ﬁle system primitives lay other features, extensions, and more sophisticated operations.

The Structure of a File is given the concept of a ﬁle, a ﬁle system may impose no structure on the ﬁle, or it may enforce a considerable amount of structure on the contents of the ﬁle. An unstructured, “raw” ﬁle, often referred to as a “stream of bytes,” literally has no structure. The ﬁle system simply records the size of the ﬁle and allows programs to read the bytes in any order or fashion that they desire. If a ﬁle system chooses to enforce a formal structure on ﬁles, it usually does so in the form of records. With the concept of records, a programmer specifies the size and format of the record, and then all I/O to that ﬁle must happen on record boundaries and be a multiple of the record length.