

File-System Interface



Practice Exercises

- 10.1 Some systems automatically delete all user files when a user logs off or a job terminates, unless the user explicitly requests that they be kept; other systems keep all files unless the user explicitly deletes them. Discuss the relative merits of each approach.
- 10.2 Why do some systems keep track of the type of a file, while others leave it to the user or simply do not implement multiple file types? Which system is “better?”
- 10.3 Similarly, some systems support many types of structures for a file’s data, while others simply support a stream of bytes. What are the advantages and disadvantages?
- 10.4 Could you simulate a multilevel directory structure with a single-level directory structure in which arbitrarily long names can be used? If your answer is yes, explain how you can do so, and contrast this scheme with the multilevel directory scheme. If your answer is no, explain what prevents your simulation’s success. How would your answer change if file names were limited to seven characters?
- 10.5 Explain the purpose of the `open()` and `close()` operations.
- 10.6 Give an example of an application in which data in a file should be accessed in the following order:
 - a. Sequentially
 - b. Randomly
- 10.7 In some systems, a subdirectory can be read and written by an authorized user, just as ordinary files can be.
 - a. Describe the protection problems that could arise.
 - b. Suggest a scheme for dealing with each of the protection problems you named in part a.

- 10.8 Consider a system that supports 5000 users. Suppose that you want to allow 4990 of these users to be able to access one file.
- a. How would you specify this protection scheme in UNIX?
 - b. Could you suggest another protection scheme that can be used more effectively for this purpose than the scheme provided by UNIX?
- 10.9 Researchers have suggested that, instead of having an access list associated with each file (specifying which users can access the file, and how), we should have a *user control list* associated with each user (specifying which files a user can access, and how). Discuss the relative merits of these two schemes.