

COS214 Tutorial 7

Roberto Togneri, 2000

1.
 - (a) The Macintosh OS allows file names of the form “Tutorial Six”, so does MS Windows 95. As the C shell with the UNIX OS uses a space as a command line separator what are the consequences of having a name with an embedded space? Could you have a UNIX filename with an end of line character in it for example?
 - (b) A company uses a mixture of UNIX workstations and Pentium PC’s running Windows 95 connected to the same network. For ease of administration user accounts are stored on a UNIX file server and all machines have access to the server across the network. A user, on one of the UNIX workstations, creates a file and names it, *ChocolateChipCookies.doc*. The same user now wishes to edit that file on a Pentium PC using MS Word. Can this be done? If not, can you suggest a cure? Does it make any difference if the Pentium is running Windows 3.1x or Windows 95? What if the file was created on one of the Pentium systems and then accessed via a UNIX workstation?

2.
 - (a) What is the purpose of a filename extension? How does UNIX know whether a file contains an executable program?
 - (b) UNIX has many different file types. What is the purpose of this? Give examples of the different types.
 - (c) List the typical attributes a files might have? Can you think of other attributes a file ought to have?

3.
 - (a) If */usr/jim* is the working directory, what is the absolute path name for the file whose relative path name is *../ast/x*?
 - (b) What are the advantages and disadvantages of recording the name of the creating program with the file’s attributes (as is done in the Macintosh operating system)?

4.
 - (a) A file can be copied in two different ways:
 - (i) Simply “rename” the file by updating the directory information
 - (ii) Copying the file to the new location and deleting the original fileCompare the two approaches. What are the corresponding UNIX commands?
 - (b) Some systems automatically open a file when it is referenced (for reading or writing) for the first time, and close the file when the job terminates. Discuss the advantages and disadvantages of this scheme as compared to the more traditional one, where the user has to open and close the file explicitly.

5.
 - (a) An operating system only supports a single directory but allows that directory to have arbitrarily many files with arbitrarily long file names. Can something approximating a hierarchical file system be simulated? How?
 - (b) Directories can be implemented as “special files” that can only be accessed in limited ways, or as ordinary data files. What are the advantages and disadvantages of each approach?

6.
 - (a) The UNIX file system supports hard and symbolic links. Is there an equivalent technique in MS Windows 95? Does this technique behave like a hard link or symbolic link?
 - (b) What additional parameters should be passed to the file create function besides the name of the file?
 - (c) It has been suggested that the first part of each Unix file be kept in the same disk block as its inode, rather than locating the inodes in the first part of the filesystem. What would the advantages be? And what are the disadvantages?