Drew Pulliam – DTP180003

Week 6 Essay

Reflection on APUE 3.9-3.12

Section 3.9 starts off with a simple program used to copy files using read/write. It then goes a bit more in depth explaining how BUFFSIZE changes how fast the copy can be completed, and it suggests a size of 4096 is usually sufficient as larger sizes have diminishing returns.

Section 3.10 describes how the kernel represents open files. The three data structures used are: 1. A process table with entries for each open process, and file descriptors inside the process entries. 2. A file table of all open files, including status flags for each file. 3. A v-node structure contains information about each file. All 3 of these tables have been part of UNIX for a long time, and they are very important regarding how files are shared with different applications on the UNIX machine.

Section 3.11 starts by explaining how appending to a file by two different processes simultaneously could create problems. If two processes open a file and determine that the last line is line 500 for example, but then process A writes 5 more lines to the file, process B might still think the end is line 500. This could result in process B overwriting process A’s data. The way UNIX solves this is by making an O\_APPEND flag. This flag will allow all processes using a file to append to that file without fear of other processes overwriting their data. This problem can also happen when two processes try to create a file with the same name, which could erase the first one created. UNIX again solves this by combining the check for files with the same name with creating a file. When processes try and do these actions separately it could create overwritten files.

Section 3.12 is about dup and dup2 functions. They are used to duplicate file descriptors. Dup simply creates the lowest number available file descriptor, and dup2 takes two arguments, which allows you to specify what number descriptor you would like to use as the duplicate.