MOSS Memory Management Simulator

Lab Work 4

1. Use mkfs to create a file system with a block size of 64 bytes and having a total of 8 blocks. How many index nodes will fit in a block? How many directory entries will fit in a block? Use dump to examine the file system backing file, and note the value in byte 64. What does this value represent? Use mkdir to create a directory (e.g., /usr), and then use dump to examine byte 64 again. What do you notice? Repeat the process of creating a directory (e.g., /bin, /lib, /var, /etc, /home, /mnt, etc.) and examining with dump. How many directories can you create before you fill up the file system? Explain why.

2. Enhance ls.java to display for each file the uid and gid as decimal numbers, and the 9 low-order bits of mode as a 3-digit octal number (i.e., 000..777).

3. Write a program find.java which, given a path name, checks to see if the path exists, and if so lists that path name and all files in all directories (and sub-directories, and sub-sub-directories, etc.) under it, one path name per line. For example:

   ```
   java find /home
   ```

   might produce the following output:

   ```
   /home
   /home/nathant
   /home/nathant/bar.txt
   /home/nathant/foo.txt
   /home/rayo
   /home/rayo/homer
   /home/rayo/homer/odyssey.txt
   /home/rayo/homer/iliad.txt
   /home/rayo/virgil
   /home/rayo/virgil/aeneid.txt
   /home/rayo/virgil/eclogues.txt
   /home/rayo/virgil/georgics.txt
   ```

   under the right circumstances, of course. Hint: Your program may include a recursive method or an array for keeping track of each directory as you open it. What is the maximum directory tree depth to which your program will work?