6.033 Handson Exercise 2

Michael Salib

February 14, 2002

1. ps -u msalib –sort=-vsize
2. ps -u msalib | wc -l
3. yes ‘yes elephant | head -4 | fmt‘ | head -10
4. ls -l –sort=size /bin/*.conf* | head -5
5. The temp file construction will produce bad data in cases where the first program continues indefinitely and the user relies on a later stage of the pipe to terminate the whole expression. Any expression involving the yes program will fail in this way.

   Besides producing incorrect results for some cases, the temp file formulation has other problems. It forces the creation of a file on disk that will always be deleted. If a filter is terminated abnormally, stale temporary files will be left.

   Another problem with the temp file construction is that it serializes execution. When using pipes, all programs in the pipe are running in parallel streaming data from one to the other. This means that later programs in the pipe can begin processing data as soon as the earlier programs start producing output. Serial execution implies that later programs can't begin working until the earlier programs have finished.
6. When using a ; to sperate commands, the output is sequenced correctly. But when I use & to seperate commands, different lines of output are interleaved. I expected that the output ordering for the first command would be determinate since the two programs are forced to execute serially. I expected that output for second command would have lines arranged in an indeterminate order because the two programs generating them were running in parrallel.

   What happened was that both invocations of myyes began at about the same time. One of them (I'll call it myyes1) was scheduled to run first. It produced the first line of output, but blocked on the file write operation. Since myyes1 was sleeping until the write returned, the kernel scheduled myyes2 to run. It produced one line of output and then blocked while writing that output to the file. By this time, myyes1's write had completed so myyes1 began running again. It wrote one line of output after which myyes2 regained control and wrote the last output line.
7. I spent one hour working on this assignement.