Computer Science 415.340

Operating systems

HOW TO DO IT

At last we have completed our analysis. From an initial requirement that people be able to do useful work with computers, we have argued our way down to the level at which we have ideas which correspond fairly directly with the computer hardware available. If they didn't, we'd have to argue further; if the computer hardware had provided much more sophisticated functions, we might have been able to stop sooner. As it is, we have expressed our wishes for the computer system in terms of two items provided by the computer – storage, and a processor – and one abstraction of the things we want to happen – processes. Now all we have to do is to work back up again, implementing the various ideas according to our designs and using the machinery available.

Our first task is therefore *to make a process work*. What does that entail ? We listed some points in the chapter *IMPLICATIONS OF PROCESSES*; these were our major conclusions as to what would be necessary in any implementation. It is interesting (if not entirely accidental) that the first three of these are about storage, and the very first introduces an implementation feature which is of great importance – though so commonplace that few people notice it.

This is the requirement that "it is necessary that the processor should have access to the programme instructions while the process is active". It is this requirement that leads to the universally accepted stratification of storage into, at the simplest level, file store and memory; further developed, it leads into file caches, processor caches, translation lookaside buffers, and many other devices which on the one hand give us the fast processing to which we have become accustomed, but on the other hand can significantly complicate the operating system's task in managing events in the computer.

We can be a little more precise. The force which leads us to develop all the wonders we listed is the interaction of the requirement mentioned at the beginning of the previous paragraph and the much more practical requirement that processing be done *quickly*. In our analysis, we have not said much about processing speed, because we've been concerned more with the principles of getting work done; now we come to the practice, it is obviously an important consideration which must be catered for in implementation. We therefore begin our study of execution by examining the practical implications of the requirement for speed, and in particular how this leads us to invent memory.