appeared in 2003

Doing many things at once

A computer’s processor has to act like a juggler, says S.Ananthanarayanan.

A juggler keeps several balls in the air with just two hands. A computer processor can handle only one job at a time, but like the juggler, it keeps many processes going!

Many processes

The computer could be playing music from a CD and running a word processing programme at the same time, for instance. And at the same time, it could be checking with the clock and calendar, to see if it were 8 am of 6th Oct, to remind you of somebody’s birthday.

The computer’s processor, maybe the Pentium chip in your computer, does not actually do all these things at the same time. It does them in rapid succession, over and over again, so that each process seems to get almost continuous attention. This is possible because each process actually does not need continuous attention of the processor. After the processor has read information from the CD, hard disk or the keyboard or even memory, processed the information and is sending the result out to a storage or display medium, the processor needs to sit idle till the next bit of information comes in.

This is usually through another read from the CD or the hard disk, which involves mechanical action, like the spinning of the CD/disk. Now, these readings from CD or disk take so many hundreds of times longer than the time the processor takes to process information, that the periods of idling of the processor are long indeed. And this idling time can be used to take care of some other process.

The ring master

It is the operating system that manages this better use of processor time. Operating systems with this capacity do not manage tasks but they manage a complicated thing called a task or process manager. As the user starts processes, one by one, the manager stops other processes, during their fleeting spells of ‘processor idleness’ and hands the processor over to the new process.

This action of ‘stopping’ and ‘handing over’ is a pretty complicated thing. A process may well be ‘stopped’ to make room for another one, but it will need to be restarted, a fraction of a second later. All the values and status at the time of ‘stopping’ then need to be stored when the process is stopped and then restored when the process is restarted.

The operating system needs to do this complicated thing hundreds of times a second, while it juggles the different processes, keeping them in the air during their periods of ‘input-output’ and letting them have the processor, usually in turn, only while they need processing.

Parallel processing

Sometimes it becomes possible to split a single job into two and get the two parts done at the same time, as two processes running at the same time. The job should then save time, as each process would make progress ‘for free’, while the other process was obtaining or writing data. For instance, searching a dictionary for a specific word. If this were the 1,29,773rd word, it would take that many tries before it was found. Each try would mean reading a word from the dictionary and making a comparison. The comparisons would be very fast, in the processor, but each ‘fetch’ of the words, even from computer memory, would take many times longer. Now, if two processes, were set to check alternate words, the processor could be comparing one word even while the next word was being fetched, and the third word could be fetched while the second was being compared and so on.

In this way, the total time is only the time of all the ‘fetches’, the processor time need not be considered at all. If the job were done with 3, 4 or even a hundred processes running at the same time, the search, or, more often, some complex operation, could become significantly faster.