SWAPPING

Peter J. Denning, Naval Postgraduate School, Monterey, California

January 2008
Revised 5/26/08

Abstract: Swapping is a general term for exchanging blocks of program code or data between main and secondary memory.

Keywords: operating systems, virtual memory, context switching

Swapping is a general term for exchanging blocks of program code or data between main and secondary memory. An executing program encounters swapping in three ways:

- Swapping in: Moving the entire program into main memory at the start of execution.
- Swapping out: Moving the entire program out of main memory at the completion of execution.
- Page swapping: Moving individual pages of the program in or out of main memory during execution of a program.

The time to complete a swap is typically 10,000 to 100,000 times the basic instruction time. Since swapping is so expensive compared with instruction execution, system designers have always sought ways to minimize and mask it.

The term originated in the time-sharing systems of the early 1960s. These systems had insufficient memory to allow more than one program to be loaded (swapped in) for execution at a time. At the end of a time slice or stop for input/output, their operating systems swapped out the executing program and then swapped in another program. A single program could be swapped many times during its execution. Swapping was a perfect description of the main work of time-sharing -- switching the CPU from one program to another.

In those early systems, swapping and CPU execution were disjoint activities. The operating system controlled the swapping overhead by setting time slices to be multiples of the average swapping time. MIT's CTSS was able to guarantee that the CPU would be executing programs about 50% of the time with this strategy (1).

To improve CPU efficiency to near 100%, operating systems of the late 1960s incorporated multiprogramming. Swapping was limited to swapping in and swapping out. The CPU could be switched among loaded programs without

further swapping. Swaps were masked by performing them in parallel with CPU execution, without interrupting or slowing the CPU.

Multiprogramming is often combined with virtual memory. In that case, the operating system may allocate fewer pages of memory to a program than the size of its address space. There will be page swapping during execution. The operating system maintains a complete copy of the address space in a swap file on the disk; each page fault swaps a page from that file with a page from main memory. The paging algorithm attempts to minimize page swapping.

Modern operating systems use swapping in all these forms. Windows XP and Vista, Mac OS 10, and Linux all combine multiprogramming and virtual memory. They allow users the option of turning off virtual memory; in which case, the operating system will swap in a program's full address space at the beginning and then execute without page swapping.

BIBLIOGRAPHY

1. F. J. Corbato, M. Merwin-Daggett, and R. C. Daley, An experimental time sharing system, *Proceedings of the Spring Joint Computer Conference 21*. In: S. Rosen (ed.), *Programming Languages and Systems*. New York, McGraw-Hill, 1967, pp335-344.

FURTHER READING

A. Tanenbaum, *Modern Operating Systems*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 2007.