CIS520 Operating Systems History

Dr. Daniel Andresen

OS development driven by relative cost

In the beginning: Expensive Hardware, Cheap People

Goal: maximize hardware utilization.

Now: Cheap Hardware, Expensive People

Goal: make it easy for people to use computer.

In the early days:

Problem: Code to manipulate external I/O devices is very complex, and is a major source of programming difficulty.

Solution: Build a subroutine library (device drivers) to manage the interaction with the I/O devices. The library is loaded into the top of memory and stays there.

Problem: computer idle during job setup

Solution: Hire a specialized person to do setup.

Solution: Build a batch monitor. Store jobs on a disk (*spooling*, have computer read them in one at a time and execute them). Debugging now offline. No more instant feedback.

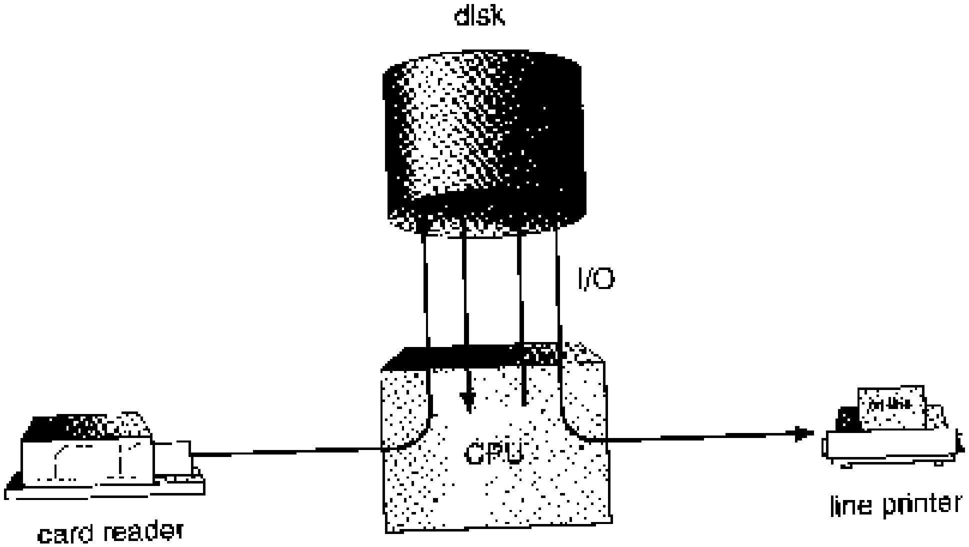


Figure 1.3 Spooling.

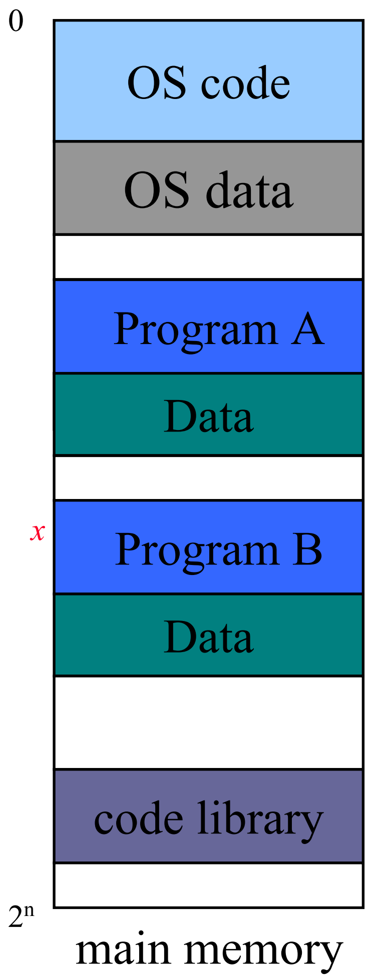
Problem: Computer idle during runtime

Problem: At any given time, job is actively using either the CPU or an I/O device, and the rest of the machine is idle.

Solution: Allow the job to overlap computation and I/O. *Buffering* and interrupt handling added to subroutine library.
(aka *asynchronous* I/O)

Problem: one job can't keep both CPU and I/O devices busy.

Problem: jobs are compute- or I/O-bound. Get poor utilization either of CPU or I/O devices.
Solution: multiprogramming - several jobs share system. Dynamically switch from one job to another when the running job does I/O.
Big issue: protection



Phase shift II: Computers become much cheaper. People costs become significant.

Issue: It becomes important to make computers easier to use and to improve the productivity of the people.

Problem: having to wait for batch output.
Solution: interactive timesharing.
Problem: batch scheduling
Solution: Preemptive scheduling.
Problem: People need to have their data.
Solution: Add file systems for quick access to data.
Problem: The boss logs in and gets terrible response time because the machine is overloaded.
Solution: Prioritized scheduling.

An example of **resource allocation** problems. The timeshared machine was full of limited resources (CPU time, disk space, physical memory space, etc.) and OS responsible for allocating of the resources.

Phase III: Computers become even cheaper. One computer to each user.

- Initial cost is very important in market.
- Minimal hardware, minimal OS.
- Protection, security less of an issue.
- OS resource consumption becomes a big issue
- OS back to a shared subroutine library.
- Hardware becomes cheaper and users more sophisticated.
- People need to share data.
- Networking and security become very important.

- OS start putting back features present in the old time sharing systems (OS/2, Windows NT, even Unix).

Rise of network.

- Internet drives new ways of thinking about computing. *Operating system is no longer interface to the lower level machine* - people structure systems to contain layers of middleware. So, a Java API may be the primary thing people need, not a set of system calls.

- Network computer - get resources off network, use middleware layer for compatibility.