AMSC/CMSC 662
Computer Organization and Programming for Scientific Computing

Fall 2011

Operating Systems

Dianne P. O'Leary

Operating Systems

Notes taken from " How Operating Systems Work" by Curt Franklin and Dave Coustan

http://computer.howstuffworks.com/operating-system.htm

- What is an operating system?
- Jobs for the OS
- Most popular OSs

What is an operating system?

An operating system is software that organizes and controls hardware and software so that the device it lives in behaves in a flexible but predictable way.

Simple devices such as microwave ovens, clocks, etc. don't have operating systems, but smartphones, computers, etc. do.

What does an operating system control?

Picture: http://computer.howstuffworks.com/operating-system2.htm

- Schedules processes for resources such as CPU access and printing.
- Ensures consistent performance, even if a program is interrupted.
- Provides a machine-independent application program interface (API) so that applications like web browsers can work on different hardware systems.
- Allocates space on disks
- Helps users with hardware updates, such as addition of a new printer.

Types of operating systems

- Real-time (RTOS): used for systems like MRI machines, flight controllers, braking systems, and robots for which prompt response to inputs is essential.
- Single user, single task: like Palm OS.
- Single user, multiple task: like Microsoft Windows.
- Multi user: like Linux.

What happens when you power-up your laptop?

Step 1: A piece of code stored in read-only memory (ROM) (why?) on the computer's motherboard (main board of computer chips) performs these tasks:

- Check out the CPU, memory, and basic input-output systems (BIOS) using a power-on self test (POST).
- Start up the basic input-output systems (BIOS) software (actually, "firmware"), also on the motherboard, to activate the disk drive(s).
- This triggers the first piece of the operating system, the bootstrap loader, stored on the main disk.

Step 2: The bootstrap loader loads the full operating system into memory.

- Sets up driver programs to control various hardware systems like disks, keyboard, screen, ports, etc.
- Partitions RAM into OS space and user space.
- Establishes data structures for flags and semaphores so that the hardware systems can communicate.
- Turns control over to OS.

Jobs for the OS

- Processor management
- Memory management
- Device management
- Storage management
- Application interface
- User interface

Processor management

- As much as possible, keep processors busy with threads/processes.
- Handle interrupts due to
  - error conditions (overflow, attempts to access memory out-of-bounds, etc.)
  - delays (e.g., page faults, wait for user input)
- Keep track of which processes are waiting for the CPU and which are suspended due to waiting for keyboard input, disk service, etc.
- Give each process a long enough time slice of CPU time to make progress without unduly delaying other processes, and prevent loss of efficiency due to thrashing, where most time is spent swapping processes.

When a process needs to relinquish CPU control the OS, (if possible) after the instruction pipeline is emptied:

- Saves (to RAM or disk) the process control block (PCB) for the process. This contains all information specifying the current state of the process:
  - An ID number that identifies the process and its priority.
  - Program counter (PC).
  - Register contents.
  - States of various flags and switches.
  - Pointers to the upper and lower bounds of the memory required for the process.
  - A list of files opened by the process and pointers to next byte in each.
  - The status of all I/O devices needed by the process.
- Restores the state of the process that will replace it using its PCB, sets a timer for when it will be interrupted, and sets the CPU to execute starting at the PC.

Picture: http://computer.howstuffworks.com/operating-system5.htm

Memory management

- Each process must have enough memory in which to execute, and it can neither run into the memory space of another process nor be run into by another process.
- The different types of memory in the system must be used properly so that each process can run most effectively.

This is done through virtual memory.

Picture: http://computer.howstuffworks.com/operating-system7.htm

Device management

Anything not on the motherboard must have a device driver to control it.

The driver takes data that the OS put in a file and translates them into streams of bits placed in specific locations on storage devices, or a series of laser pulses in a printer.

Each device driver can be a separate process that must be scheduled by the OS.

Provide APIs

Most programs call upon the operating system to perform tasks such as I/O. For example, the C functions `fopen` and `fprint` call upon the operating systems, and so do the corresponding MATLAB functions.

Applications rely on these interfaces to be reliable and consistent across various hardware systems.

This allows systems such as MATLAB acroread, firefox, etc. to be built and distributed.

Provide user interfaces

User interfaces such as window systems (e.g., gnome) and graphical user interfaces (GUIs) are built on top of operating systems to make life easier.

Unix shell systems such as the Bourne shell and the C shell are also user interfaces.

Most popular OSs

- Microsoft Windows variants.
- Mac OS X.
- Linux variants.