Operating Systems

Introduction
Based on
OS Concepts by SGG

What an OS does

- From a User point of view
  - Provides facilities to make the computer usable
    - Disk, network, screen, etc, support
    - Security, connectivity, flexibility
  - Allows many programs and users to share the computer
- From a System point of view
  - Resource allocator (time, memory, bandwidth...)

What **IS** an OS

- Many slightly varying definitions
- Usually the kernel and the system programs
    - Kernel is the software that manipulates the hardware directly (e.g. instruct the disk head to move)
    - System programs is the software that handles the basic system operations (e.g. initialize the wifi)

OS Operation

- Many programs normally run on an OS
  - Regular programs
  - Daemons
- User programs run in user mode
- When they want an OS service execute a software interrupt
  - A software interrupt is similar to a hardware interrupt.
- Monitor mode is not the same as running as root.

Software interrupt

- When an interrupt is issued
  - The CPU stops what it is doing and attends the interrupt
  - The CPU executes code in the interrupt vector
  - The code in the interrupt vector is trusted and executes in monitor (or system) mode
  - When the code finishes the CPU continues execution

System Calls

- System calls are implemented with software interrupts, aka traps.

- To the user look like regular function calls but inside
  - They put the arguments to the system call to the right place (in registers, in blocks of memory, or on the stack)
  - They issue a system call
  - Clean up and propagate the return value or the error messages.

Types of system calls

- Process control (fork, start, stop, wait, etc)
- File management (open, close, remove, read, write, etc)
- Device management (open, close, read, write, control, memory map, etc)
- Information maintainance (time, date, system data, file or device information, process information)
- Communication (start, end communication lines, send receive messages, control)

Memory Structure

- There is a hierarchy in memory
  - Registers
  - Cache
  - RAM
  - SSD
  - Hard disk

I/O

- The CPU communicates with the I/O device through the bus

- The communication involves interrupts and exchange of small chunks of information (the CPU reads/writes from/to the device or writes/reads to/form the memory)

- It is faster if the device writes directly to the memory (DMA: direct memory access)

- The s/w that orchestrates this is called device driver.

Resource Management

- Processes (and threads)
- File system
- Mass storage
- Cache
- I/O

Single-, Multi-Processor Systems

- In the past most systems were single processor

- Now most are multiprocessors (multicore)

- The advantages include:
  - Increased performance
  - Economy of scale
  - Increased reliability

- The main reason for the popularity of such systems is that it is increasingly too expensive to design/make/run uniprocessors of equivalent performance.

Multprocessing Jargon

- SMP: Symmetric Multiprocessing

- UMA: Uniform Memory Access

- NUMA: Non-UMA

- Multicore: many CPUs on a single chip

- Hyperthreaded: can run two programs at the same time on a single core

- Cluster Systems: many computers connected via a fast network.

Multiprocessing Jargon

- Client Server
- Peer-to-peer
- Virtualization
- Cloud Computing
    - Private cloud
    - Public cloud
    - SaaS: software as a service
    - PaaS: platform as a service

Virtualization

- Technology that allows one CPU to run multiple OSes

- Usually there is a host OS that runs normally.

- An application VMM (Virtual Machine Manager) runs a guest OS.

- Very popular among developers, testers. Also used to run servers

Real-Time Systems

- Systems with time critical performance
  - Sound and video recording

- Embedded systems
  - IoT: internet of things, so that the refrigerator can keep ordering broccoli as fast as you can throw it into the garbage.

Open Source Systems

- Open source movement is several decades old.
- In 1983 Richard Stallman started GNU
- In 1991 Linux Torvalds started the Linux kernel
- In 2003 Android was founded (and bought by Google in 2005)
- There are other open source Oses: BSD, Solaris(?)