**OPERATING SYSTEMS LEC 01**

20th April – Lecture in tuition free week

**Operating System**

An OS is a program that acts as an interface between a user and the computer hardware. It provides a basis for running applications.

**Kernel**

Is a program that is running all the time on a computer and provides services to the system and its users. System programs are run at boot time. When running they are called system process or system daemons.

Once booted it waits for events to happen and probes the hardware buses to see what is present. These are called interrupts and are received from hardware through the system bus.

In Linux the first system process is called **init**. It runs other daemons.

**Purposes**

1. Provide an environment for a user to execute programs conveniently and efficiently
2. Simulate features not available on hardware
   1. Virtual machines, virtual memory, etc.
3. Control resources and provide a basis which applications can be written

**Computer**

* Has one or more CPU
  + Brain of system
* Memory system
* Input/output devices eg. Disks, printers

A system bus connects all of these together.

Each device has a controller in charge of a particular device type (eg. Disks) and each has a local buffer.

**Bootstrap**

An initial computer program that runs when powered up. It is stored in hardware called firmware such as ROM. Initialises all aspects of a system such as CPU registers. Knows how to locate the OS kernel and run it.

**Interrupts**

When an interrupt occurs, the CPU stops its current task and jumps to a specific location to execute the **interrupt service routine (ISR)**. It then continues its task once the interrupt is completed (which is saved before the ISR is carried out).

Lower number – higher priority.

During I/O various devices raise interruptions to indicate certain events (eg. that they have completed output). Also used for exceptions in programs.

**Speed**

Registers - > cache - > main memory (all these 3 are volatile)

Solid state drives - > magnetic drives - > optical drives - > magnetic tapes (non-volatile)

**CPU**

Executes instructions only from memory. Uses main memory (RAM) which is the only large storage media that it can access directly.

Execution cycle

1. Fetches an instruction from memory and stores it in a register.
2. Decodes the instruction
3. Fetch any operands (eg. X +y… get x)
4. Execute and store the results in memory
5. Repeat the steps

**I/O Concepts**

Any I/O device communicates with a system by sending signals over a cable or wireless via a port. A device controller is a collection of electronics that operate a port, a bus or a device. One or more devices may use a common set of wires called a bus.

**Host controller interactions**

Uses:

* Polling
* Interrupts

**Basic interrupt I/O Mechanism**

1. Device controller raises and interrupt by asserting a signal to the CPU
2. CPU catches the interrupt and dispatches the interrupt to the ISR
3. Interrupt handler clears the interrupt and resumes with interrupted task.

**Direct Memory access**

Bypasses CPU to transfer data directly between I/O device and memory to improve the performance of large data transfer. Allows CPU to go on with other work concurrently.

CPU gives command to DMA controller, sending pointer with sender and receiver address and also the number of bytes to be transferred. Once finished the DMA interrupts the CPU.

**Multiprocessor Systems**

* Increased throughput – gets more work done in less time
  + Not by n times due to overhead
* Economy of scale – I/O devices, memory and power supplies can be shared.
* Increased reliability – failure of one processor does not make the system down.

Asymmetric – One master processor which schedules and allocates work to slave processors. Each slave processor waits for instructions from its masters.

SMP (symmetric) – each processor runs an identical copy of the OS and has its own registers and cache allowing for many processes to un.

Multicore systems have multiple processor but on the same chip. These are more efficient because on-chip communication is faster and inter-chip communication.

A clustered system consists of multiple CPUs.

A Beowulf cluster consists of multiple computers interlinked by LAN.

**Multiprogramming**

Is multiple tasks stores in memory at the same time with the CPU multiplexed among them.