Computer Science & Engineering Department
I. I. T. Kharagpur

Operating System: CS33007

3rd Year CSE: 5th Semester (*Autumn 2006 - 2007*)
Lecture I (**Introduction**)

Goutam Biswas *Date:* 24th-25th July, 2006

- An OS is
  1. a software(!)
  2. an interface between the hardware (instruction set architecture (ISA)) and the application software [application programimming interface (API[1]) and application binary interface (ABI[2])],
  3. a resource manager - manages resources among different users and computations,
  4. a creator of virtual machines - creates an illusion of more resources than what is physically present.
- A hardware interface or ISA consists of
  1. the instruction set of the machine,
  2. accessible CPU, FPU and MMU registers,
  3. other visible features of CPU, FPU and MMU e.g. interrupt structure, DMA, exceptions, memory management etc.,
  4. I/O address mapping (separate I/O space or memory mapped I/O space) etc.
- An OS interface hides the hardware details and provides a more friendly view through the system calls[3]. System calls have special CPU instruction called software interrupt or trap.
- External and internal views of an OS:
  1. The external view of an OS is through system calls that are used (called) by an application programmer (API). Many of the calls are platform independent e.g. POSIX[4] [Portable Operating System Interface (X is for Unix)]. At a lower level application binary interface (ABI) defines interface for compiled application programs e.g. machine instructions, data representations, related data structures. This depends on the hardware platform and also the OS.
  2. The internal view of an OS is the structure and implementation of the interface on top of the instruction set and system architecture.
- Following are some of the resources managed by the OS

[1] See API in Wikipedia, the free encyclopedia: `http://en.wikipedia.org/wiki/Main_Page`

[2] See ABI in Wikipedia.

[3] See system call in Wikipedia.

[4] See POSIX Wikipedia.

1. CPU+FPU+MMU - given to different threads of computations (time multiplexed).
2. Main memory - partitioned and given to different process images and data. One physical partition may also by used by different objects at different time.
3. Disk - partitioned into different file systems. Under each file system there may be different users, and each user may have multiple files on different partitions.
4. I/O devices - shared by different processes e.g. printer, disk.
5. Data - different user may access data (may be with some restriction) for communication between processes.

- Resource management calls for protection of resources of one computation from another. OS uses architectural support of the processor to protect computation and its resources.

- Several computations can run on a single computing system. OS creates virtual machines for each computation. A computation under an OS that sees a virtual machine is called a process.

  A process or the virtual machine context in which the computation takes place has several components that are mapped to the physical resources. The components are - CPU, code or text, data, stack, open files, signal received from the OS (exceptions, interrupts etc. are translated to signals by the OS) etc. The CPU is mapped to the CPU; code+data+stack are mapped in the memory; open files are related to the files or devices.

  More than one threads of computation may share parts of a virtual machine e.g. code, data, open files etc. But other virtual components e.g. CPU, stack may be different for them. These are called threads.

- A virtual CPU -

  1. OS creates a virtual CPU+FPU+MMU for a process by providing a subset of instructions of the physical processor. In fact the facility is provided by the processor itself, and is used by the OS for management.
  2. Instructions to control I/O, memory and CPU state are not available to a user process as the OS does the resource management.
  3. OS provides ‘pseudo instructions’, for some of the prohibited instructions, through system-calls. A computation requests for service from OS through these calls.

- Virtual memory -

  1. Every process image (executable module) is created in a logical memory space. It is also called virtual memory. This space is often divided into different regions e.g. code or text (read-only), data (read-write), stack (read-write, but grows in a different way). The logical memory space is mapped to the physical memory when the process image is loaded. This calls for address translation from the logical space to the physical space.

- Virtual I/O -

1. OS hides the complexity of programming an I/O device controller and provides simple system call interface to the application program. The system-call invokes the appropriate device drivers that gives command to the device controller.
2. I/O devices are treated as files in Unix or Linux, and a process with proper capability can access these files using the system-calls appropriate to files e.g. read, write etc.

- Different types of system calls -

    1. I/O operations - creation of file, read, write
    2. request for more memory - increase of data area, shared memory area
    3. creation of a new virtual machine - process creation, thread creation
    4. communication and synchronization between computations
    5. and many more