Operating Systems Concepts

Introduction and Overview

What is an operating system?

Top-down view

- Provides an extended or virtual machine abstraction to user programs
- **Easier to program than the underlying hardware.**

  All services are invoked and accomplished through system calls.

Bottom-up view

- Acts as a resource manager of a complex system
- Resources consist of processors, memories, timers, disks, mice, keyboard, network interfaces, printers etc.
- OS manages allocation of these resources to user programs in an orderly and controlled manner

Resource multiplexing

- **OS multiplexes resources in two ways:**

  In time, In space

  Time multiplexing involves different programs taking turns in using the resource. Example: CPU scheduling, printer sharing.

  Space multiplexing involves different program getting part of the resource possibly at the same time. Example: memory is divided into several running programs.

The major OS issues

- Structure: how is the OS organized?
- Sharing: how are resources shared across users?
- Naming: how are resources named (by users or programs)?
- Security: how is the integrity of the OS and its resources ensured?
- Protection: how is one user/program protected from another?
- Performance: how do we make it all go fast?
- Reliability: what happens if something goes wrong
- (either with hardware or with program)?
- Extensibility: can we add new features?
- Communication: how do programs exchange information,
- including across a network?

More OS issues

Concurrency: how are parallel activates (computation and I/O created and controlled?

Scale: what happens as demands or resources increase?

Persistence: how do you make data last longer than program executions?

Distribution: how do multiple computers interact with each other?

Accounting: how do we keep track of resources usage, and perhaps charge for it?

Protection and security as an example

- None
- OS from my program
- Your program from my program
- My program from my program
- Access by intruding individuals
- Access by intruding programs
- Denial of service
- Distributed denial of service
- Spoofing
- Spam
- Worms
- Viruses
- Stuff you download and run knowingly (bugs, Trojan horses)
- Stuff you download and run unknowingly (cookies, spyware)

Type of Operating Systems

- **Main frame operating systems**

  Huge amounts of I/O activity. 1000s of disks not unusual

  Provide batch, transaction and time sharing services.

  Batch processing is routine non-interactive jobs. e.g. claims processing, sales reporting etc.

  Transaction processing systems handle large number of small requests e.g. check processing in banks, air line reservations etc.

  Time-sharing systems allow multiple remote users to run jobs at the same time e.g. querying a database.

  OS Optimized for these tasks. Example is OS/390

- **Server operating systems**

  Run on very large PCs, workstations or even main-frames. They serve multiple users over a network simultaneously and allow them to share hardware and software. Examples are web servers, database transaction servers etc. Examples of OS in this class are Win2K, XP and flavors of UNIX.

- **Multiprocessor operating systems**

  OS is basically a variation to server operating systems with special provisions for connectivity and communication management between different CPUs.

- **PC operating systems**

  OS provides a nice interface to a single user. Typically used for word processing, spread sheets, Internet access etc.

Real-time operating systems

Characterized by time as the key parameter. Real-time response to internal and external events is more important than any other design goal. Classified into two sub-categories: Hard and Soft real-time

Example applications include Industrial process control, robotics and assembly lines, air traffic control, network routers and telecommunication switches, multi-media systems etc.

Embedded operating systems

Embedded in small devices e.g. palm-top computers e.g. PDA, TV sets, micro-wave ovens, mobile phones. They have characteristics of real-time systems (mainly soft real-time) but also have restraints on power consumption, memory usage etc. Examples include PalmOS and Windows CE. Height of this type is smart-card systems.

Distributed Systems

Distribute the computation among several physical processors.

Loosely coupled system – each processor has its own local memory; processors communicate with one another through various communications lines, such as high-speed buses or telephone lines.

Advantages of distributed systems

- Resources Sharing
- Computation speed up – load sharing
- Reliability
- Communications

Parallel Systems

Multiprocessor systems with more than on CPU in close communication.

Tightly coupled system – processors share memory and a clock; communication usually takes place through the shared memory.

Advantages of parallel system:

- Increased throughput
- Economical
- Increased reliability
- graceful degradation
- fail-soft systems