IT Fundamentals

Dr/Mai Ramadan Ibraheem

Lecturer at Information Technology Dept.,

Faculty of Computers and Information – KFS University

Outline

- Introduction.
- Machine level representation of data.
- Digital logic.
- Assembly level machine organization
- Hardware realizations of algorithms.
- Operating systems and virtual machines.
- Computing applications.
- Introduction to net-centric computing.

Outline

- Virtual Machines
- Operating Systems

Virtual Machines

- A virtual machine (VM) is an operating system (OS) or application environment that is installed on software, which imitates dedicated hardware.
- The end user has the same experience on a virtual machine as they would have on dedicated hardware.
- Virtual machines allow you to run other operating systems within your current operating system –as if they're just another program on your computer.

A System

- A set of principles and concepts and all of the components which obey these principles.

An Operating System

- A system of software which manages the hardware resources of the system to provide a base for its users' programmatic computing needs.
- The operating system acts as an interface or intermediary between the user of a computer and the computer hardware.
- Other definition are below.

Processes

- The OS supports operations on processes: Create a process; Delete a process; Suspend a process; Resume a process; Inter-process communication; Inter-process synchronization; Create/delete a subprocess.
- OS allocate memory space for programs; de-allocate memory space when needed; maintain the mappings from virtual to physical memory; decide how much memory to allocate to each process, and when a process should be removed from memory (policies).

Parts of an operating system page 108

- Kernel
- Device Drivers
- User Interface
- System Utilities
- Shell

Kernel

- handles:
- Loading / Unloading applications from memory
- Scheduling tasks to run on the CPU
- Memory management
- File management
- Data security

Device Drivers

- Every piece of hardware that makes up the computer or connected to it, will have a device driver that allows the operating system to control and communicate with it.
- Hundreds of device drivers pre-installed with the operating system, and the right ones for that particular computer set-up is **loaded on boot-up.**
- A device driver for Windows is **different** from the device driver for Linux.

User interface

- Graphical User Interface (GUI)
- directing what you see on the screen (via the device driver) and reacting to your key presses and other inputs.

System Utilities

- the basic facilities that run in the background without user interaction. For example, Print spool services; Cryptographic; password management; File management services.

Shell

- The interface to the operating system.
- The outermost layer of the operating system. Incorporate a programming language to control processes and files,
- allowing communication with the operating system via a control language, letting the user control the peripherals without knowing the characteristics of the hardware used.

EVOLUTION OF OPERATING SYSTEMS

- Serial Processing
- Simple Batch Systems
- Multiprogrammed Batch Systems
- Time-Sharing Systems
- Personal-Computer Systems (PCs)
- Parallel Systems
- Asymmetric multiprocessing model
- Distributed Systems
- Hard real-time systems

Serial Processing

- The programmer interacted directly with computer hardware i.e., no operating system.
- These machines were run with a **console** consisting of display lights, and a printer.
- If an **error occur** indicated by **lights**. Programmers examine the registers and main memory to determine error. If the program is success, then **output** will appear on the **printer**.

Simple Batch Systems

- To speed up processing, jobs with similar needs are batched together and run as a group.

Multiprogrammed Batch Systems

- Jobs run sequentially, on a first-come, first-served basis.
- However when several jobs are on a direct-access device, job scheduling is run as multiprogramming to increases CPU utilization.

Time-Sharing Systems

- or **multitasking** is a logical extension of **multiprogramming**. That is **processors time** is shared among multiple users simultaneously.
- The main difference between Multiprogrammed Batch Systems and Time-Sharing Systems is in Multiprogrammed batch systems its objective is **maximize** processor use, whereas in Time-Sharing Systems its objective is **minimize** response time.

Personal-Computer Systems (PCs)

- is dedicated to a single user.
- Micro computers are considerably smaller and less expensive than mainframe computers.
- For e.g., MS-DOS, Microsoft Windows and Apple Macintosh.

Parallel Systems

- Multiprocessor systems have more than one processor.
- The advantages :
- Throughput (Number of jobs to finish in a time period)
- Save money by sharing peripherals.
- Increase reliability
- Fault-tolerant (Failure of one processor will not halt the system).
- Symmetric multiprocessing model: Windows 7 and Windows Server 2008.
- Each processor runs an identical job (copy) of the operating system, and these copies communicate.

Asymmetric multiprocessing model

- Each processor is assigned a **specific task**.
- A master processor controls the system.
- Sun's operating system SunOS version 4 is a asymmetric model.

Distributed Systems

- Distributed systems distribute computation among several processors.
- Instead, each processor has its own local memory.

Real-time Systems

- are used when there are **rigid time requirements** on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application.
- has well-defined, fixed time constraints.
- E.g., Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, and home-applicance controllers.

Real-time Systems types

- **Hard real-time systems.**
- Hard real-time systems guarantee that critical tasks complete on time.
- **Soft real-time systems.**
- Soft real time systems are less restrictive. Critical real-time task gets priority over other tasks and retains the priority until it completes. e.g., Multimedia, virtual reality.

EVOLUTION OF OPERATING SYSTEMS

- Serial Processing
- Simple Batch Systems
- Multiprogrammed Batch Systems
- Time-Sharing Systems
- Personal-Computer Systems (PCs)
- Parallel Systems
- Asymmetric multiprocessing model
- Distributed Systems
- Hard real-time systems

Operating Systems

- the core of any modern technological advancement.
- Operating Systems enable other programmers to do their job easier, as they do all the low-level operations such as interfacing with the hardware.


End


Thank You