Computer Organisation
Operating Systems

Edward Cheung
email: icec@polyu.edu.hk
24 July, 2003.

Operating Systems

- Services and facilities provided by OS
- Single-job operations
- Concurrent operations
  - Multitasking
  - Multiprocessing
- OS organization & architecture
  - Monolithic kernel
  - Hierarchical kernel
  - Microkernel
- Types of user interface
- bootstrapping

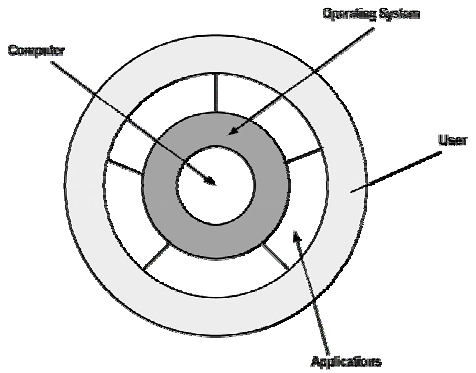
Agenda

- Hierarchical view of a computer and the function of an OS
- Evolution and development of OS
- Uniprogramming & Multiprogramming Systems
- Time Sharing System & Real Time OS
- Multiprocessor Systems & Clustered Systems
- Practical Desktop OSs
- Structure of OS, monolithic, layer structure, virtual machine, microkernel
- Single Job and Concurrent Systems
- Process, Threads, Context Switching
- I/O access, file system
- Bootstrapping and User Interface
- Windows 2000 example

Components of a Computer System

- Hardware – provides basic computing resources; for example, CPU, storage, I/O devices.
- Operating system – controls and coordinates the use of the hardware among the various application programs for the various users.
- Utilities – Software Programs designed for system management and software development; for example, account management, program development.
- Applications – Software Programs designed for user to make use of system resources to solve computing problems; for example, scientific computing, database, computer aided design, simulation, games, business applications.

Hierarchical View of a Computer



Four Main Functions of an OS

- From the user point of view, an OS should provide the following functions:-
  - Managing hardware
    - Enable user to utilize computer hardware efficiently.
    - Resolves conflicts for resource contention.
  - Managing files
    - Organise information in an efficient manner.
  - Providing a user interface
    - Enable user to use the computer conveniently.
  - Managing application
    - Control the execution of computer program

Services and facilities provided by OS

- An Operating System is a program that acts as an interface between a user of a computer and the computer hardware.
- It is the most important program of a computer because it manage all the computer software and hardware.
- In addition to the basic functions on resources management, modern OS provide a security framework to protect system information and resources.
  - For example, access to I/O, computational resources and files.
- Modern OSs are bundle with utilities for system application and management.
  - Text editors, compilers, linkers, debuggers, etc

Some Considerations in Operating System

- Must adapt to hardware upgrades and new types of hardware. Examples:
  - Character vs. graphic terminals, pen input – tablet PC
  - Wireless devices, InfraRed devices, WLAN
- Must ready to offer new services,
  - e.g., internet support, multimedia, communication
- Must offer flexibility and support software reuse:-
  - modular construction with clean interfaces.
  - object oriented methodology.
- OS has two groups of users; the programmer and the ordinary user.

Views of an Operating System

- Traditionally, an OS can be view as:-
  1. Resource Manager – manages and allocates resources.
  2. Control program – controls the execution of user programs and operations of I/O devices.
  3. Command Executer – Provides an environment for running user commands.
- Modern Operating System is a virtual machine:
  - An interface between the user and hardware that hides the details of the hardware (e.g. I/O).
  - Constructs higher-level (virtual) resources out of lower-level (physical) resources (e.g. files).

Classification, Evolution and History of OS

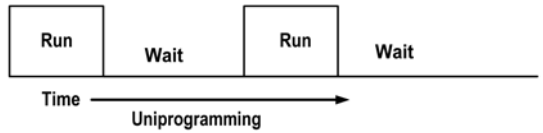
- Early Systems (1950)
- Simple Batch Systems (1960)
- Multiprogramming Batch Job Systems (1970)
- Time-Sharing and Real-Time Systems (1970)
- Personal/Desktop Systems (1980)
- Multiprocessor Systems (1980)
- Networked/Distributed Systems (1980)
- Handheld Systems (1990)
- Wireless Systems (2000)

Lesson from Simple Batch Systems

- Alternate execution between user program and the monitor program.
- Automatic Job Sequencing was used to transfer control from another job on job completion
- Deck of cards was used for input on one line per card. Control cards and Job Control Languages (e.g. $LOAD, $DATA, $END, etc.) are used to identify different applications and data. e.g. Fortran, load data, etc.
- A technique called SPOOLing: Simultaneous Peripheral Operation On Line was used to overcome slow peripherals like line printer, tape drives. This allow I/O and CPU to overlap.
- A resident monitor is used to control the system – the OS.

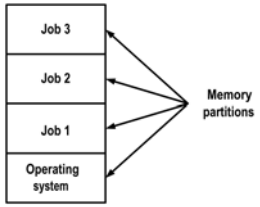
Problem in Uniprogramming

- I/O operations are exceedingly slow (compared to instruction execution).
- A program contains even a very small number of I/O operations will spend most of the system time waiting for I/O completion.
- Poor CPU usage when because only one program is allow in the memory.

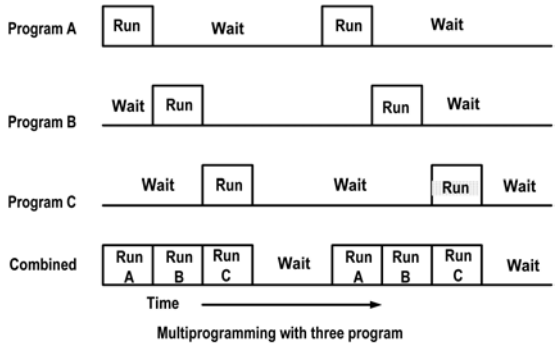


Multiprogramming Batch Job System

- Allows the CPU to execute another program while a program must wait for an I/O device.
- The size of the memory must be large enough to hold several programs in a multiprogramming environment.
- The OS is more complex since it has to manipulate computing resources.



A Multiprogramming System in Action



Time Sharing Systems (TSS)

- Batch job multiprogramming environment does not support interaction with users.
- TSS extends the batch job multiprogramming environment to handle multiple interactive jobs.
- TSS provides an interactive multiprogramming environment to enable multiple users simultaneously access the system through terminals.
- Processor's time is shared among multiple users.
- Because of slow human reaction time, a typical user needs 1-2 second to interact with the system. The idle system can be utilised by other user.
- The OS is more complicate because a job may be swapped in and out of memory to disk.

Real-Time Operating Systems (RTOS)

- RTOS are dedicated systems that need to adhere to deadlines , i.e., time constraints.
- Correctness of the computation depends not only on the logical result but also on the time at which the results are produced.
- Hard Real-time System – *Jobs must meet their deadlines*
  - This criteria conflicts with general purpose time-sharing systems which targeted at the best utilization of computing resources.
  - Often appear as control device for dedicated application:-
    - Industrial control, Robotics, Life Maintenance
  - Limited resources for better control
    - e.g. small size memory, limited type, program in ROM, etc.
      - http://www.ucos-ii.com
- Soft Real-time System – *Meeting Deadline is desirable but not mandatory*
  - Useful in modern applications (multimedia, virtual reality) requiring advanced operating-system features.

Desktop Systems

- Personal computer is a computer system dedicated to a single user or a small number of users.
- I/O devices – manipulated many different resources; keyboards, mouse, display, multimedia components, communication adapters, etc.
- The design goal is targeted at user convenience, attractive UI, robust but at low cost.
- Full features to support consumer type products.
- The OS must ready to adopt new technology/peripherals.
- Rapid program development environment

Multiprocessor Systems

- Asymmetric multiprocessing
  - master processor schedules and allocates work to slave processors.
- Symmetric multiprocessing (SMP)
  - Each processor runs an identical copy of the operating system.
  - Typically each processor does self-scheduling form the pool of available process.
  - Most modern operating systems support SMP.
  - e.g.Windows NT/2000/.NET, Solaris, etc.

Symmetric Multiprocessing (SMP)

- Each processor can perform same functions and share the same memory space and I/O facilities (symmetric).
- The OS schedule processes/threads across all the processors (real parallelism).
- Existence of multiple processors is transparent to the user.
- Incremental growth in processing power by adding CPUs.
- Robustness: a single CPU failure does not halt the system, only the performance is reduced.

Clustered Systems

- Clustering allows two or more systems to share external storage and balance CPU load.
- Asymmetric clustering
  - one server runs the application while other servers standby.
- Symmetric clustering
  - all N hosts are running the application.
- Closely coupled system
  - processors also have their own external memory.
  - communication takes place through high-speed channels.
  - Provides high reliability.

Networked/Distributed Systems

- Loosely coupled system
  - each processor has its own local memory.
  - processors communicate with one another through various communications lines.
- Applications
  - Resources Sharing
  - Load sharing
  - Reliability
- Network Operating System (NOS)
  - provides mainly file sharing.
  - Each computer runs independently from other computers on the network.
- Distributed Operating System (DOS)
  - gives the impression there is a single operating system controlling the network.
  - network is mostly transparent – it's a powerful virtual machine.

Practical Desktop Oriented OS

- DOS
  - DOS is one of the legacy but popular example of OS.
  - Windows 3.x GUI have been developed running on top of OS
  - It is a good troubleshooting tool because it is simple and small.
- Windows 9x
  - This included Window 95/98/Me
  - This was the most popular OS. Still using DOS core.
- Windows NT/2000/XP
  - Many different versions tailored for different applications; Workstation:-
    - Professional Edition / Home Edition, Advanced Server
  - Server:-
    - Advanced Server/ Enterprise Server / Datacenter Server to name a few

Practical Desktop Oriented OS (cont.)

- UNIX
  - Lack of consistency from one vendor's version to another.
- Linux
  - Free basic OS, difficult to install Use X-Windows
- Mac OS X
  - Mach 3.0 microkernel and 4.4 BSD-Lite2 kernel
  - Darwin Project – http://www.opendarwin.org
  - Finder to provide the desktop GUI
  - Aqua is Apple's name for the GUI
- Function as both server and workstation depending on configurations
- As majority of the computer is on IA-32 architecture, we shall bias towards Microsoft Windows System in our discussion.

Operating System Structure

- During normal operation of a computer system, a portion of the operating system remains in main memory to provide essential services to the system. This portion of the OS is known as the kernel.
- 5 different structures
  - Monolithic System
  - Layered System
  - Virtual Machine
  - Microkernel
  - Client-Server Model

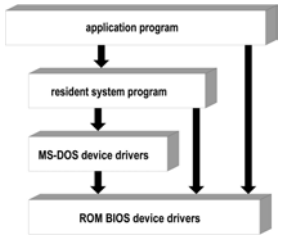
Monolithic Systems

- No structure at all
- Normally provide the most functionality in minimum space – small kernel

Examples:-

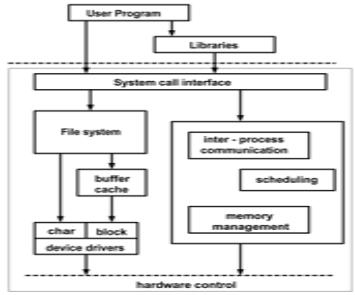
•MS-DOS

Although MS-DOS has some structure, its interfaces and levels of functionality are not well separated.



•UNIX

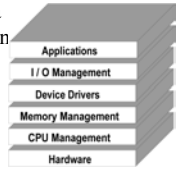
UNIX System Architecture

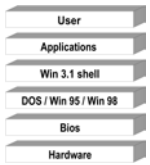


Hardware is beneath or surround by the operating-system. The heart of the operating system is called the kernel and it comes with a number of user services and interfaces such as C compiler and command shell.

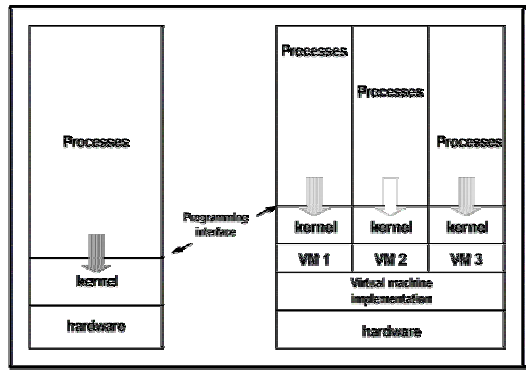
Layered Systems

- The operating system is divided into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface.
- Layers are selected with modularity concept. Each level only relies on the function or service of the lower layers for more primitive function.
- This approach decomposes a large problem into a number of more manageable problems – divide and conquer approach.





Non-virtual vs. Virtual Machine



IBM VM/370

VM/370 is an OS with virtual machines simulated on System/370. It provide multiple users with seemingly separate and independent S/370 system.

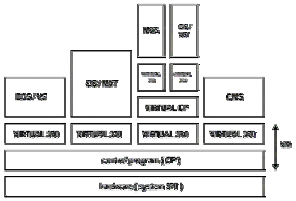
DOS/VS – Disk Operating System / Virtual Storage

OS/VS1 – Operating System / Virtual Storage 1

OS/MVT – Multiprogramming with a Variable number of Task

MVS – Multiple Virtual Storage

CMS – Conversational Monitor System



Ref:- Creasy, R.J., "The Origin of the VM/370 Time-Sharing System", IBM J. R&D Vol 25 Issue 5, 1981.

Current System Ref. :- http://www.ibm.com/servers/eserver/zseries/

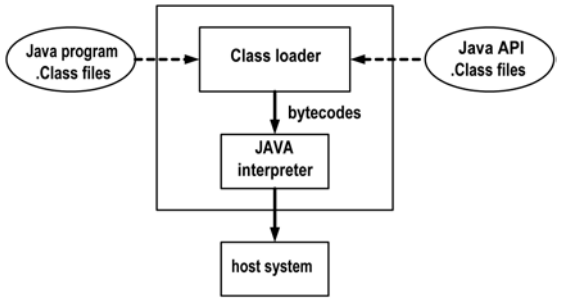
Advantages/Disadvantages of Virtual Machines

- The virtual-machine concept provides complete protection of system resources since each virtual machine is isolated from all other virtual machines. This isolation permits no direct sharing of resources.
- A virtual-machine system is a perfect vehicle for operating-systems research and development. System development is done on the virtual machine, instead of on a physical machine and so does not disrupt normal system operation.
- The virtual machine concept is difficult to implement due to the effort required to provide an *exact* duplicate to the underlying machine.

Java Virtual Machine

- Compiled Java programs are platform-neutral bytecodes executed by a Java Virtual Machine (JVM).
- JVM consists of:
  - class loader
  - class verifier
  - runtime interpreter
- Just-In-Time (JIT) compilers increase performance.

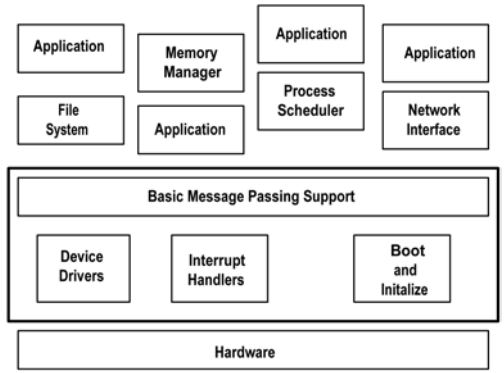
The Java Virtual Machine



Microkernel System

- Move as much functionality as possible from the kernel into *"user space"*.
- Only a few essential functions remain in the kernel
  - primitive memory management (address space)
  - I/O and interrupt management
  - Inter-Process Communication (IPC)
  - basic scheduling
- Other OS services are provided by processes running in user mode
  - device drivers, file system, virtual memory
- Inherent distributed system support since communication takes place between user modules (machines) using message passing.
- A performance penalty is caused by replacing service calls with message exchanges between process.

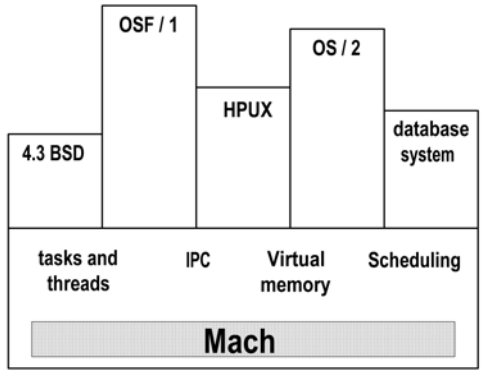
Microkernel Architecture



Advantage of a Microkernel Architecture

- more flexible, extensible, portable and reliable
  - modular design is extensible and it is more easy to add services
  - small microkernel simplified testing
  - changes needed to port the system to a new processor is done in the microkernel - not in the other services.
- Object-oriented operating system
  - components are objects with clearly defined interfaces that can be connected to software
- Examples of microkernels are Mach and QNX
  - http://www.qnx.com

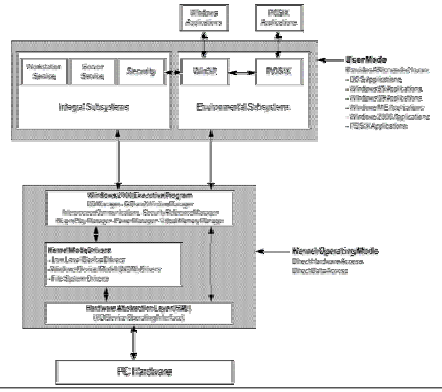
Mach 3 Microkernel Structure



Windows NT

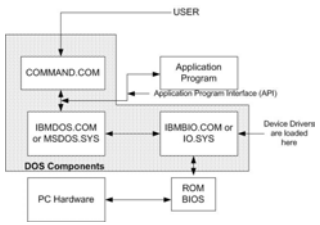
- Windows NT is a "modified microkernel" – the modularity of microkernel design but implement the modules in kernel mode for performance.
- many of the system functions outside the microkernel run in kernel mode
- Uses a layered structure
  - Hardware abstraction layer (HAL)
  - makes hardware look the same to the kernel
  - provides support for symmetric multiprocessing

Microsoft Windows Application Architecture



Single-job operations

- Single-tasking
  - OS can run only one program at a time, passing a single thread to the CPU.
  - Example is Microsoft DOS



Multitasking

- Multitasking
  - OS is doing more than one thing at the same time. OS managed to share resources between different tasks.
- Cooperative Multitasking
  - Not a true multitasking category, Also known as task switching
  - In this category, if the other task is busy as visible by the hourglass on the screen, the user cannot immediately switch to another process like click or select another windows on the desktop.
  - Example is Windows 3.x
- Preemptive Multitasking

Concurrent System

- Some systems allow execution of only one process at a time (e.g., early personal computers).
  - uniprogramming system
- Some systems allow more than one process to execute at a time or concurrent execution of many processes.
  - multiprogramming systems
- In a multiprogramming system, the CPU switches automatically from one process to another with each process running for a period of time. In reality, the CPU is actually running one and only one process at a time.

Concurrent System (cont.)

- The computation process is organized into a number of (sequential) processes, each viewed as a block of code with a pointer showing the next instruction to be executed.
- The system need a fair scheduling mechanism
  - each process gets a chance to run
- The system must protect the process from interfere with each other and they do not modify each other's state

Classification of Concurrent Operations

- hardware parallelism:
  - CPU computing, one or more I/O devices are running at the same time.
- pseudo parallelism:
  - rapid switching back and forth of the CPU among processes, pretending that those processes run concurrently.
- real parallelism:
  - can only be achieved by multiple CPUs. Single CPU systems cannot achieve real parallelism, but keeping track of multiple activities is difficult.

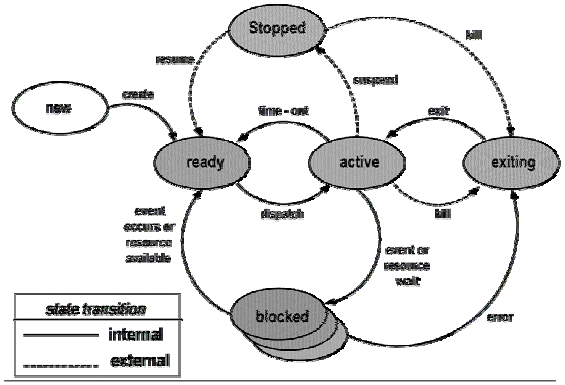
Process

- A process is basically a program in execution.
- A process has its own address space
  - Address space is a list of memory location that the program can read and write and it contains
    - Executable code
    - Data
    - stack
    - Pointers and registers such as program counter, stack pointer, etc.
- A process table (An array of structures) is used to save the information of each process.
- A process can create child processes – process tree.
- Communication between different process is called interprocess communication.

Process states

- A process can has a number of *states*.
- the operation of a multiprogramming system can be described by a state transition diagram on the process states. The states of a process include:
  - **New**—a process being created but not yet included in the pool of executable processes - resource acquisition
  - **Ready**—processes that are prepared to execute when given the opportunity.
  - **Active**—the process that is currently being executed by the CPU.
  - **Blocked**—a process that cannot execute until a event occur
  - **Stopped**—a special case of **blocked** where the process is suspended by the operator or the user.
  - **Exiting**—a process that is about to be removed from the pool of executable processes - resource release

Process State Diagram



Context Switching

- When an event occurs, the operating system saves the state of the *active* process and restores the state of the *interrupt service routine (ISR)*. This mechanism is called a **Context Switch**.
- Everything that the next process could or will damage must be saved.

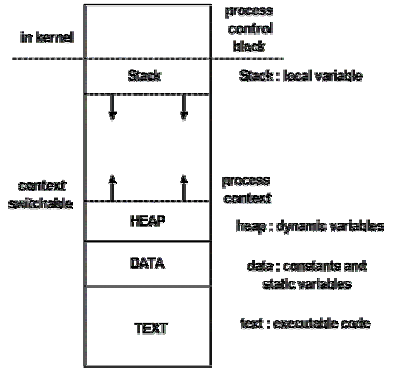
  For example:
  - Program counter (PC)
  - Program status word (PSW)
  - File access pointer(s)
  - Memory
- While saving the state, the operating system should mask (disable) all interrupts.
- The system is idle during context switching. Hence, context switch time is a overhead of the process.

Process Control Block

- The operating system must know specific information about processes in order to manage and control them.
- Such information is usually grouped into two categories:
  - process state information
    - E.g., CPU registers (general purpose and special purpose), program counter.
  - process control information

    E.g., scheduling priority, resources held, access privileges, memory allocated, accounting.
- This collection of process information is kept in and access through a *process control block (PCB)*.
- OS dependent.

A Process Context



Threads

- Threads are light weight process (LWP)
- A *thread* is the abstraction of a unit of execution.
- As a basic unit of CPU utilization, a thread consists of an instruction pointer (also referred to as the PC or instruction counter), CPU register set and a stack.
- A thread shares its code and data, as well as system resources and other OS related information, with its peer group (other threads of the same process).

Thread vs. Process

- A thread operates in much the same way as a process:
  - can be one of the several states;
  - executes sequentially (within a process and shares the CPU);
  - can issue system calls.
  - Creating a thread is less expensive.
- Switching threads within a process is cheaper than switching between threads of different processes.
- Threads within a process share resources (including the same memory address space) conveniently and efficiently compared to separate processes.
- Threads within a process are NOT independent and are NOT protected against each other.

Accessing OS Services

- A system call is used to get access to the OS. This is a privileged operation.
- System call interface is a set of functions that can be used by the user to access system services.
- The only difference between a “procedure call” and a “system call” is that a system call changes the execution mode of the CPU to a privileged mode.

Examples of System Call (UNIX)

- Process control
  - fork(), exec(), wait(), abort()
- File manipulation
  - chmod(), link(), stat(), creat()
- Device manipulation
  - open(), close(), ioctl(), select()
- Information maintenance
  - time(), acct(), gettimeofday()
- Communications
  - socket(), accept(), send(), recv()

Interprocess Communication (IPC)

- Methods for effective sharing of information among cooperating processes are collectively known as *interprocess communication (IPC)*.
- Two basic methods are used:-
  - shared memory
    - Data are shared among processes and they are directly available to each process in their address spaces.
  - message passing
    - Data are explicitly exchanged among processes.

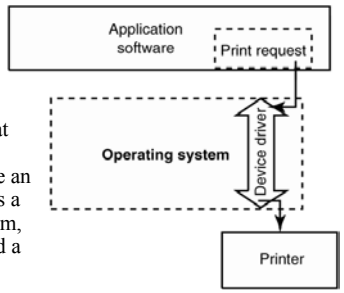
Message Passing System

- A common approach in communication is where one process *sends* some information to another. The information exchanged among processes in this way is called a *message*.
- A message can be a structured (language) object, specified by its type, or it is specified by its size: fixed length or variable length.
- There are two basic operations on messages:
  - **send()**—transmission of a message.
  - **receive()**—receipt of a message.
- The OS component which implements these operations is called a ‘‘ *message passing*’’ system.

I/O Device Access

Device driver

- A software module that attaches to the OS and allows programs to use an external device such as a printer, a display system, or a disk drive is called a device driver.



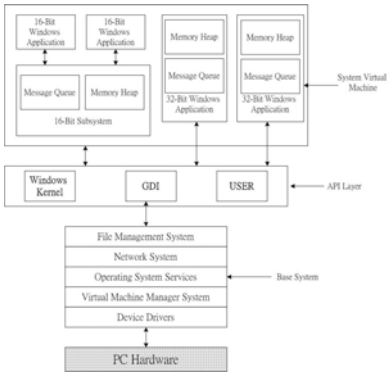
Microsoft OS - Real and Protected Modes

- Real mode is 16-bit data bus
  - Single task
  - Legacy Windows
  - Max. 1024k bytes memory
  - A memory extender for OS to provide addressing space larger than 1MB
  - For compatibility only
- Protected mode is 32-bit data bus
  - Multitasking
  - Windows 98 / NT
  - Max 4096M bytes memory (limited by Motherboard)
  - Does not allow direct access to RAM but instead, the OS is equipped with Virtual memory, swap file, page file, etc.

Real Mode and Protected Mode (cont.)

- Real mode is still supported by today's CPU and OS.
- Windows 9x / NT /2000 / XP start out in real mode and then switch to protected mode
- 32-bit OS can run 16-bit programs and 16-bit OS cannot run 32-bit programs.
- Windows XP's support for Itanium-based Systems
  - Microsoft Win64 API and Win32 API
  - Virtual memory up to 16 terabytes (TB)
  - Interoperable with IA-32 architecture
  - Non-paged pool up to 128GB (256MB in Win32)

Windows 9x System Structure



Windows NT Modes

- User Mode
  - Programs have only limited access to system information and can access hardware only through other OS services.
  - Example is the NT Virtual DOS Machine (NTVDM)
- Kernel Mode
  - Programs can access the hardware layer directly
  - All drivers are kernel mode programs

File Systems

- Information stored in files must be persistent
- Not affected by process creation and termination
- A file should only disappear if the owner explicitly remove it.
- Files are managed by the OS
  - Structure of the file
  - Naming rules
  - Access:- read/write
  - Protection
  - Implementation

Loading of an OS - Bootstrapping

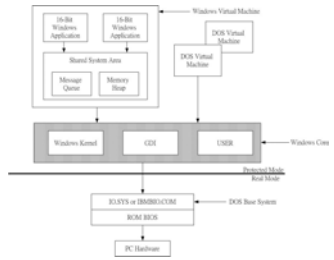
- A technique of bootstrapping is used in initializing the computer and loading the operating system. - boot
- A small program that resides in the non-volatile memory of the computer is being loaded first and the small program will command the loading of the OS which can be resided in the hard disk or in a on-line server.
- e,.g. Boot Sequence of Windows 2000
  - POST – motherboard
  - Boot initialization – OS detection
  - Bootstrap loading
  - Ntldr
  - Boot.ini
  - Ntdetect.com

Command Interpreter

- Shell
  - In operating systems such as Unix and its variants, the shell is a layer of the OS that sits between the user and the kernel. The user interacts only with the shell.
  - command.com is the command interpreter for MS DOS
- For general computing, once the system has been loaded into the computer or the system has been boot up successfully, an active shell will allow communication with user.

Types of User Interface

- Command Line Driven Interface
  - Simple
  - Difficult for ordinary user
- Menu Driven Interface
  - For small system such as embedded controller
- Graphical User Interface
  - Computational intensive
  - Event oriented



Example:- Windows 2000

- Features
- Design Principles
- System Architecture
- System Components
- Environmental Subsystems
- File system

Features of Windows 2000

- 32-bit preemptive multitasking operating system for Intel IA-32 microprocessors.
- Key goals for the system:
  - portability
  - security
  - POSIX compliance
  - multiprocessor support
  - extensibility
  - multiple languages support
  - compatibility with MS-DOS and MS-Windows applications.
- with a micro-kernel architecture.
- Available in four versions, Professional, Server, Advanced Server, Datacenter Server.

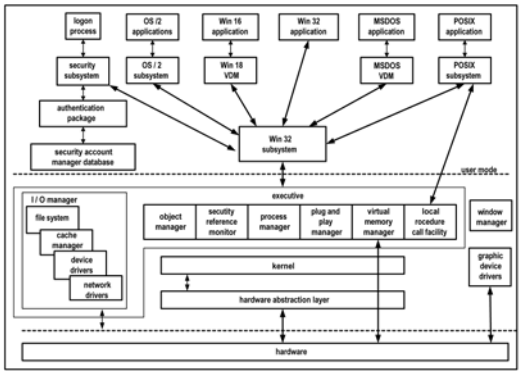
Design Principles

- Extensibility — layered architecture.
  - Executive, which runs in protected mode, provides the basic system services.
  - On top of the executive, several server subsystems operate in user mode.
  - Modular structure allows additional environmental subsystems to be added without affecting the executive.
- Portability — 2000 can be moved from one hardware architecture to another with relatively few changes.
  - Written in C and C++.
  - Processor-dependent code is isolated in a dynamic link library (DLL) called the "hardware abstraction layer" (HAL).

Design Principles (Cont.)

- Reliability — 2000 uses hardware protection for virtual memory, and software protection mechanisms for operating system resources.
- Compatibility — applications that follow the IEEE 1003.1 (POSIX) standard can be complied to run on 2000 without changing the source code.
- Performance — 2000 subsystems can communicate with one another via high-performance message passing.
  - Preemption of low priority threads enables the system to respond quickly to external events.
  - Designed for symmetrical multiprocessing.
- International support — supports different locales via the national language support (NLS) API.

Windows 2000 Architecture



Windows 2000 Architecture

- Layered system of modules.
- Protected mode — 3 main layers are:-
  - Hardware Abstraction Layer (HAL), kernel, executive.
- User mode — collection of subsystems
  - Environmental subsystems emulate different operating systems.
  - Protection subsystems provide security functions

System Files

Filename	Components
NTOSKRNL.EXE	Executive and Kernel
HAL.DLL	Hardware Abstraction Layer
WIN32K.SYS	Kernel-mode part of the Win32 subsystem
NTDLL.DLL	Internal support functions / system service dispatch stubs to executive functions
KERNEL32.DLL, ADVAPI32.DLL, USER32.DLL, GDI32.DLL	Core Win32 subsystem DLLs Export Win32 Entry Points

System Components — Kernel

- Foundation for the executive and the subsystems.
- Never paged out of memory; execution is never preempted.
- Four main responsibilities:
  - thread scheduling
  - interrupt and exception handling
  - low-level processor synchronization
  - recovery after a power failure
- Kernel is object-oriented, uses two sets of objects.
  - *dispatcher objects* control dispatching and synchronization (events, mutants, mutexes, semaphores, threads and timers).
  - *control objects* (asynchronous procedure calls, interrupts, power notify, power status, process and profile objects.)

Kernel — Process and Threads

- The process has a virtual memory address space, information (such as a base priority), and an affinity for one or more processors.
- Threads are the unit of execution scheduled by the kernel's dispatcher.
- Each thread has its own state, including a priority, processor affinity, and accounting information.
- A thread can be one of six states: ready, standby, running, waiting, transition, and terminated.

Kernel — Scheduling

- The dispatcher uses a 32-level priority scheme to determine the order of thread execution with two priority classes.
  - Real-time class thread priority from 16 to 32.
  - Variable class thread priority from 0 to 15.
- Characteristics of W2000's priority strategy.
  - Trends to give very good response times to interactive threads that are using the mouse and windows.
  - Enables I/O-bound threads to keep the I/O devices busy.
  - Complete-bound threads soak up the spare CPU cycles in the background.
- Scheduling can occur when a thread enters the ready or wait state, when a thread terminates, or when an application changes a thread's priority or processor affinity.
- Real-time threads are given preferential access to the CPU; but W2000 does not guarantee that a real-time thread will start to execute within any particular time limit – not a hard RTS.

Kernel — Trap Handling

- The kernel provides trap handling when exceptions and interrupts are generated by hardware of software.
- Exceptions that cannot be handled by the trap handler are handled by the kernel's *exception dispatcher*.
- The interrupt dispatcher in the kernel handles interrupts by calling either an interrupt service routine (such as in a device driver) or an internal kernel routine.

Executive

- Provides a set of services that all environmental subsystems can use.
- The services can be grouped as follows:-
  - Object manager
  - Virtual-memory manager
  - Process manager
  - Local-procedure call facility
  - I/O manager
  - Security reference monitor

Executive — Object Manager

- 2000 uses objects for all its services and entities; the object manger supervises the use of all the objects.
  - Generates an object *handle*
  - Checks security.
  - Keeps track of which processes are using each object.
- Objects are manipulated by a standard set of methods, namely `create, open, close, delete, query name, parse` and s`ecurity`.

Executive — Virtual Memory Manager

- The VM manager in 2000 uses a page-based management scheme with a page size of 4 kByte.
- The 2000 VM manager uses a two step process to allocate memory.
  - The first step reserves a portion of the process's address space.
  - The second step commits the allocation by assigning space in the 2000 paging file.
- A page can be in one of six states: valid, zeroed, free standby, modified and bad.

Executive — Process Manager

- Provides services for creating, deleting, and using threads and processes.
- Issues such as parent/child relationships or process hierarchies are left to the particular environmental subsystem that owns the process.

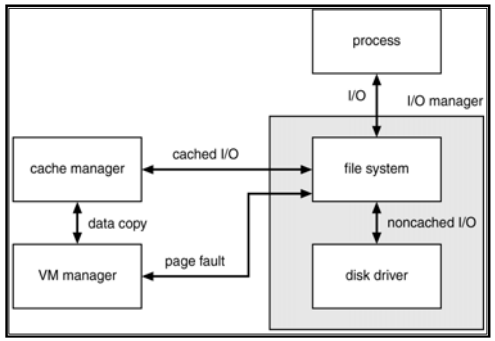
Executive — Local Procedure Call Facility

- The LPC passes requests and results between client and server processes within a single machine.
- In particular, it is used to request services from the various 2000 subsystems.
- When a LPC channel is created, one of three types of message passing techniques must be specified.
  - First type is suitable for small messages, up to 256 bytes; port's message queue is used as intermediate storage, and the messages are copied from one process to the other.
  - Second type avoids copying large messages by pointing to a shared memory section object created for the channel.
  - Third method, called *quick* LPC was used by graphical display portions of the Win32 subsystem.

Executive — I/O Manager

- The I/O manager is responsible for
  - file systems
    - Keeps track of which installable file systems are loaded, and manages buffers for I/O requests.
  - cache management
    - Controls the 2000 cache manager, which handles caching for the entire I/O system.
  - device drivers
    - Works with VM Manager to provide memory-mapped file I/O.
  - network drivers
    - Supports both synchronous and asynchronous operations, provides time outs for drivers, and has mechanisms for one driver to call another.

File I/O



Executive — Security Reference Manager

- The object-oriented nature of 2000 enables the use of a uniform mechanism to perform runtime access validation and audit checks for every entity in the system.
- Whenever a process opens a handle to an object, the security reference monitor checks the process's security token and the object's access control list to see whether the process has the necessary rights.
- Executive - Plug-and-play Manager
  - Both the device and the OS must support PnP
  - Assign interrupts and I/O memory ranges automatically.

Environmental Subsystems

- User-mode processes layered over the native W2000 executive services to enable W2000 to run programs developed for other operating system.
  - e.g. 16-bit windows, MS-DOS, POSIX,
- W2000 uses the Win32 subsystem as the main operating environment to start all processes. It also provides all the keyboard, mouse and graphical display capabilities.
- MS-DOS environment is provided by a Win32 application called the *virtual dos machine* (VDM) - a user-mode process
- Provide routines to emulate MS-DOS ROM, virtual device drivers for I/O ports
- Any MS-DOS application that needs to access hardware directly will failed to run in W2000.

File System

- The fundamental structure of the 2000 file system (NTFS) is a *volume*.
  - Created by the 2000 disk administrator utility.
  - Based on a logical disk partition.
  - May occupy a portions of a disk, an entire disk, or span across several disks.
- All *metadata*, such as information about the volume, is stored in a regular file.
- NTFS uses *clusters* as the underlying unit of disk allocation.
  - A cluster is a number of disk sectors that is a power of two.
  - Because the cluster size is smaller than for the 16-bit FAT file system, the amount of internal fragmentation is reduced.

File System — Internal Layout

- NTFS uses *logical cluster numbers* (LCNs) as disk addresses.
- A file in NTFS is not a simple byte stream, as in MS-DOS or UNIX, rather, it is a structured object consisting of *attributes*.
- Every file in NTFS is described by one or more records in an array stored in a special file called the Master File Table (MFT).
- Each file on an NTFS volume has a unique ID called a *file reference. It c*an be used to perform internal consistency checks.
- The NTFS name space is organized by a hierarchy of directories; the *index root* contains the top level of the B+ tree.

File System — Recovery

- All file system data structure updates are performed inside transactions.
  - Before a data structure is altered, the transaction writes a log record that contains redo and undo information.
  - After the data structure has been changed, a commit record is written to the log to signify that the transaction succeeded.
  - After a crash, the file system data structures can be restored to a consistent state by processing the log records.
  - This scheme does not guarantee that all the user file data can be recovered after a crash, just that the file system data structures (the metadata files) are undamaged and reflect some consistent state prior to the crash.
- The logging functionality is provided by the 2000 *log file service*.

File System — Security

- Security of an NTFS volume is derived from the 2000 object model.
- Each file object has a security descriptor attribute stored in this MFT record.
- This attribute contains the access token of the owner of the file, and an access control list that states the access privileges that are granted to each user that has access to the file.

File System — Compression

- To compress a file, NTFS divides the file's data into *compression units*, which are blocks of 16 contiguous clusters.
- For sparse files, NTFS uses another technique to save space.
  - Clusters that contain all zeros are not actually allocated or stored on disk.
  - Instead, gaps are left in the sequence of virtual cluster numbers stored in the MFT entry for the file.
  - When reading a file, if a gap in the virtual cluster numbers is found, NTFS just zero-fills that portion of the caller's buffer.

Portable Operating System Interface (POSIX)

- Different versions of UNIX such as AT&T's System V and BSD's derivatives such as Sun Microsystems' Solaris and Apple's Darwin.
- IEEE Portable Application Standards Committee (PASC) – an IEEE standard
  - http://www.pasc.org/
- POSIX compliance is required in many US government procurements.
- A POSIX compliant OS will work effectively with any application that limits itself to the POSIX standard set of Application Programming Interface (API).
- For the most part POSIX used only those functions that were common to the two mainstream UNIX variants (System V and BSD) and it is a subset of real world UNIX.
- Windows NT/2000 also support POSIX.
- The goal of POSIX was to "shrink wrapped" applications that would work without modification on different OSs.
- But when Software vendors doing development on different OSs, they usually have to tune their application in order to get the best possible performance for a particular platform. Hence compliance with POSIX doesn't mean that you have a good software.

Reference

1. A. Silberschatz, P. B. Galvin, and G. Gagne, "Operating Systems Concepts", 6th Edition, John Wiley & Sons, 2002.
   http://www.bell-labs.com/topic/books/os-book/
2. A. Silberschatz, P. B. Galvin, and G. Gagne, "Applied Operating Systems Concepts", 1st Edition, John Wiley & Sons, 2000.
   http://www.bell-labs.com/topic/books/aos-book/
3. W. Stallings, "Operating Systems: Internals and Design Principles", 4th Edition, Prentice-Hall, 2000.
   http://WilliamStallings.com/OS4e.html
4. D.A. Solomon, and M.E. Russinovich, "Inside Microsoft Windows 2000", 3rd Edition, Microsoft Press, 2000.