Topics: Modern Structure of Operating Systems
Layered Approach
- **Microkernel**
- **Modules**

UNIX
Windows NT, 2000, XP (check case study)

<u>System Structure</u>

Simple Structure:

The system structure is limited by the hardware. In the beginning MS-DOS and UNIX had a **simple structure.**

The interfaces and levels of functionality were not well separated.

MS-DOS

```
 |-----------------------------------|
|       application program           |
|-------------------------------------|
      ||                            ||
 |-----------------------------|    ||
 |   resident system program   |    ||
 |-----------------------------|    ||
      ||                  ||        ||
 |----------------------| ||        ||
 | MS-DOS device drivers | ||       ||
 |----------------------| ||        ||
      ||                  ||        ||
|-------------------------------------|
|   ROM BIOS device drivers           |
|-------------------------------------|
```

Fig. MS-DOS simple structure

MS-DOS - application programs are able to access the basic I/O routines.
This makes MS-DOS vulnerable to errant programs.

Unix

The original UNIX was limited to two separable parts: the kernel and the system programs. The main idea was that the kernel represented a whole indivisible part with a huge responsibility.

(the users)		
shells and commands compilers and interpreters system libraries		
system-call interface to the kernel		
signals terminal handling character I/O system terminal drivers	file system swapping block I/O system disk and tape drivers	CPU scheduling page replacement demand paging virtual memory
kernel interface to the hardware		

Fig. Unix Simple Structure

Modern Structure

Layered Structure:
If we have the hardware support, the OS can be broken into smaller pieces, creating a modular operating
system.
The OS is broken into a number of layers (levels), each built on top of lower layers.
The lowest (0 layer) is the hardware. The highest (N_{th} layer) is the user interface.
Each layer uses functions and services of lower-level layers only.

The new-layered structure replaces the traditional vertical architecture with a horizontal one.

WINDOWS
New terms:

Microkernel
The microkernel concept is introduced in **Windows NT** architecture.
Microkernel represents the most used and fundamental component of the OS. The microkernel functions
as a message exchange: validates messages, passes them between components, grants access to the
hardware, and performs a protection function.

HAL (Hardware Abstraction Layer)
HAL isolates the OS from platform specific hardware differences. Provides the support for Symmetric
Multiprocessing (SMP). Most of the upper level modules can access the hardware only through the HAL.

KERNEL MODE
Microkernel - most used code of the OS.
Executive

USER MODE
Environmental subsystems

User-mode processes that enable Windows to run programs developed for other operating systems:
Win32, POSIX, OS/2.
The most important subsystem is **Win32**. Win32API represents the native environment for Windows NT, Windows 2000, and Windows XP.

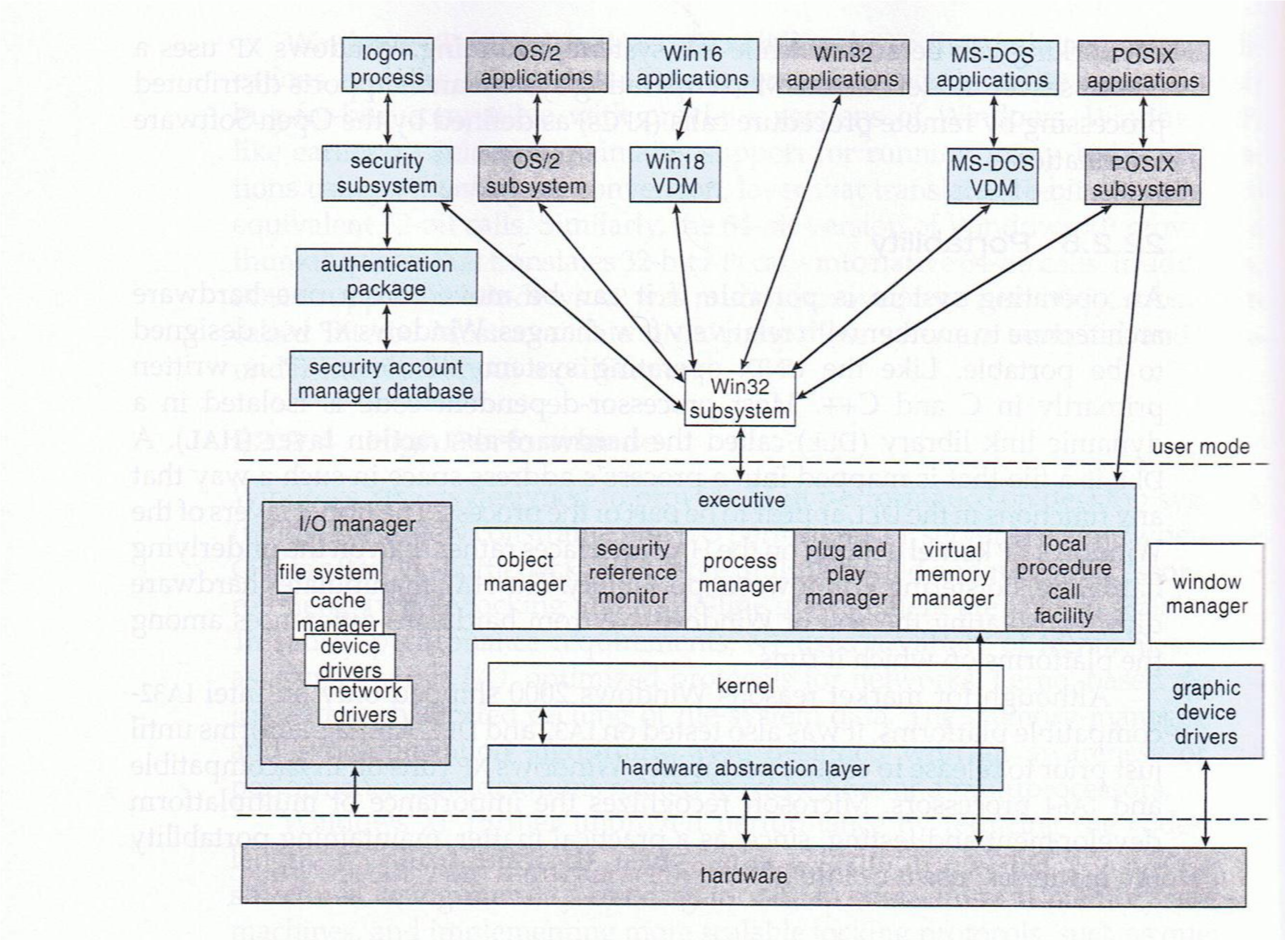

Figure 22.1 Windows XP block diagram.

UNIX
System V Release 4 (SVR4)
Features:

- Real-time processing support
- Process scheduling classes
- Virtual memory management
- Virtual file system
- Preemptive kernel

Unix runs on machines ranging from 32-bit microprocessors up to supercomputers.
A small core of facilities provides functions and services needed by a number of system processes. Each of the outer circles represents functions that can be implemented in many different ways.

Topics: Process Control Block
Process States
Process transitions and State Diagram

Process is a program in execution. A program by itself is not a process. A program can be seen as a passive entity, such the content of a file, a process can be seen as an active entity.

Process Control Block

The PCB of a process provides to the OS all key information about the process. The PCB defines a process to the operating system. The access to the PCB of a process is done using a unique process ID.

Some of the information contained by the PCB:
Process State
Program Counter
CPU scheduling information (priority, pointers to scheduling queues)
Memory-management information (base and limit register, page tables, segment tables etc.)
Accounting Information
I/O status information (list of I/O devices allocated, list of open files etc.)
CPU registers (AC, PC, general-purpose registers) content

PROCESS CONTROL BLOCK INFORMATION

Process Identification

Identifiers
Numeric identifiers that may be stored with the process control block include

- Identifier of this process
- Identifier of the process that created this process (parent process)
- User identifier

Processor State Information

User-Visible Registers
A user-visible register is one that may be referenced by means of the machine language that the processor executes. Typically, there are from 8 to 32 of these registers, although some RISC implementations have over 100.
Control and Status Registers
These are a variety of processor registers that are employed to control the operation of the processor. These include

- *Program counter:* contains the address of the next instruction to be fetched
- *Condition codes:* result of the most recent arithmetic or logical operations (e.g., sign, zero, carry, equal, and overflow)
- *Status information:* includes interrupt enabled/disabled flags, execution mode.

Stack Pointers
Each process has one or more last-in, first-out (LIFO) system stacks associated with it. A stack is used to store parameters and calling addresses for procedure and system calls. The stack pointer points to the top of the stack.

Process Control Information

Scheduling and State Information
This is information that is needed by the operating system to perform its scheduling function.
Typical items of information include

- *Process state:* defines the readiness of the process to be scheduled for execution (e.g., running, ready, waiting, halted)
- *Priority:* One or more fields may be used to describe the scheduling priority of the process. In some systems, several values are required (e.g., default, current, highest allowable).
- *Scheduling-related information:* this will depend on the scheduling algorithm used. Examples are the amount of time that the process has been waiting and the amount of time that the process executed the last time it was running.
- *Event:* identity of event the process is awaiting before it can be resumed.

Data Structuring
A process may be linked to other process in a queue, ring, or some other structure. For example, all processes in a waiting state for a particular priority level may be linked in a queue. A process may exhibit a parent-child (creator-created)
relationship with another process. The process control block may contain pointers to other processes to support these structures.
Inter-process Communication
Various flags, signals, and messages may be associated with communication between two independent processes. Some or all of this information may be maintained in the process control block.
Process Privileges
Processes are granted privileges in terms of the memory that may be accessed and the types of instructions that may be executed. In addition, privileges may apply to the use of system utilities and services.
Memory Management
This section may include pointers to segment and/or page tables that describe the virtual memory assigned to this process.
Resource Ownership and Utilization resources controlled by the process may be indicated such as opened fi1es. A history of utilization of the processor or other resources may also be included; this information may be need by the scheduler.