Windows Processes

- Implemented as objects
- An executable process may contain one or more threads
- Both processes and thread objects have built-in synchronization capabilities

Figure 4.12 A Windows Process and Its Resources
Windows Process Object

Object Type
- Process ID
- Security Descriptor
- Base priority
- Default processor affinity
- Quiet limits
- Execution time
- I/O counters
- VM operation counters
- Exception debugging flags
- Exit status

Object Body Attributes

Processes
- Create process
- Open process
- Query process information
- Set process information
- Current process
- Terminate process

Services

Windows Thread Object

Object Type
- Thread ID
- Thread context
- Dynamic priority
- Basic priority
- Thread processor affinity
- Thread creation time
- Alert status
- Suspension count
- Impersonation tokens
- Termination port
- Thread exit values

Object Body Attributes

Threads
- Create thread
- Open thread
- Query thread information
- Set thread information
- Current thread
- Terminate thread
- Get context
- Set context
- Suspend
- Resume
- Abort thread
- Test thread alert
- Register termination port

Services

(a) Process object

(b) Thread object
Windows 2000 Thread States

- Ready
- Standby
- Running
- Waiting
- Transition
- Terminated
Solaris

- Process includes the user’s address space, stack, and process control block
- User-level threads
- Lightweight processes (LWP)
- Kernel threads
Solaris Lightweight Data Structure

- Identifier
- Priority
- Signal mask
- Saved values of user-level registers
- Kernel stack
- Resource usage and profiling data
- Pointer to the corresponding kernel thread
- Pointer to the process structure
Linux Task Data Structure

- State
- Scheduling information
  - normal or real-time, priorities
- Identifiers
- Interprocess communication
- Links
- Times and timers
- File system
- Address space
- Processor-specific context
Linux States of a Process

- Running
- Interruptable
- Uninterruptable
- Stopped
- Zombie

Figure 4.18 Linux Process/Thread Model