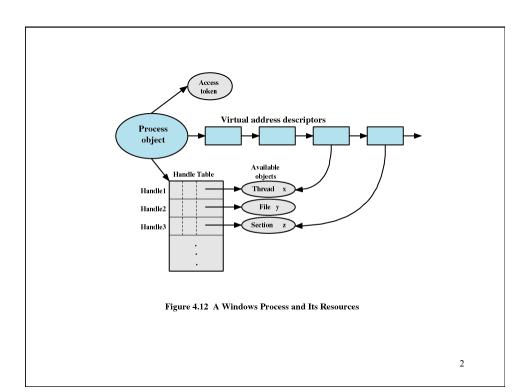
## Windows Processes

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



## Windows Process Object

Process Object Type Process ID Security Descriptor Base priority
Default processor affinity
Quota limits Object Body Attributes Execution time I/O counters VM operation counters Exception/debugging ports Exit status Create process Open process
Query process information
Set process information Services Current process Terminate process

(a) Process object

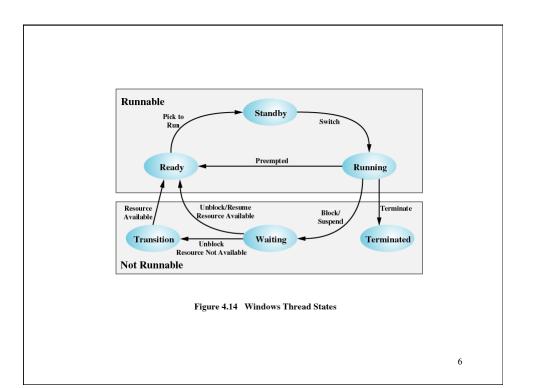
3

# Windows Thread Object

Object Type Thread ID Thread context Thread context
Dynamic priority
Base priority
Thread processor affinity
Thread execution time
Alert status
Suspension count
Impersonation token
Termination port Object Body Attributes Termination port Thread exit status Create thread Open thread Query thread information Set thread information Current thread Terminate thread Get context Services Set context Suspend Resume Alert thread Test thread alert Register termination port (b) Thread object

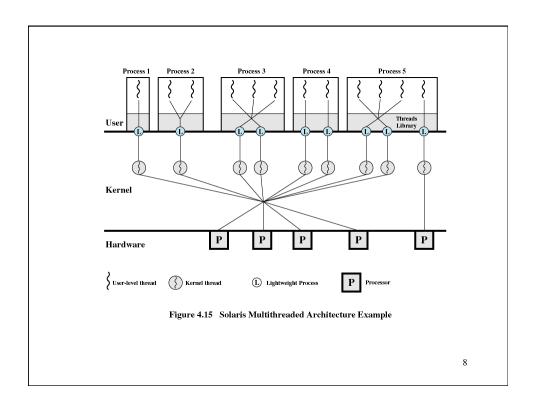
## Windows 2000 Thread States

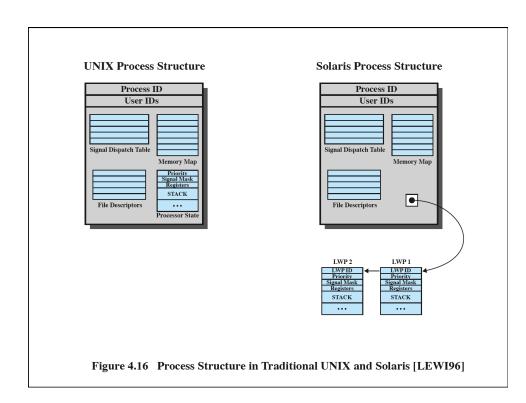
- Ready
- Standby
- Running
- Waiting
- Transition
- Terminated



## Solaris

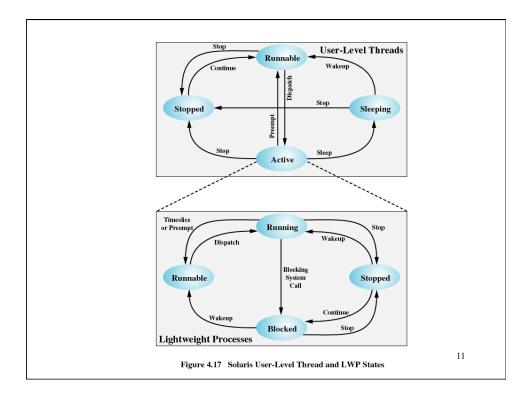
- Process includes the user's address space, stack, and process control block
- User-level threads
- Lightweight processes (LWP)
- Kernel threads





#### 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

