LSN 6
Windows Processes & Threads

ECT362 Operating Systems
LSN 6 – Windows Processes and Threads

- Windows kernel creates abstractions of threads and objects on top of the hardware
  - The Executive uses the kernel abstractions to create the model of a process in which a thread can execute
- Programs in execution are able to use the system’s resources
  - Windows divorces processes and threads
    - Processes define the logical environment in which computation can take place and the set of resources
    - Threads perform work within the process’s address space
- Windows allows for concurrent execution of threads
LSN 6 – Windows Processes and Threads

**Object Type**

**Process**
- Process ID
- Security Descriptor
- Base priority
- Default processor affinity
- Quota limits
- Execution time
- I/O counters
- VM operation counters
- Exception/debugging ports
- Exit status

**Object Body Attributes**
- Create process
- Open process
- Query process information
- Set process information
- Current process
- Terminate process

**Thread**
- Thread ID
- Thread context
- Dynamic priority
- Base priority
- Thread processor affinity
- Thread execution time
- Alert status
- Suspension count
- Impersonation token
- Termination port
- Thread exit status

**Object Body Attributes**
- Create thread
- Open thread
- Query thread information
- Set thread information
- Current thread
- Terminate thread
- Get context
- Set context
- Suspend
- Resume
- Alert thread
- Test thread alert
- Register termination port

**Services**

sschneider@udayton.edu

ECT 362 Operating Systems
LSN 6 – Process and Thread Descriptors

- Process manager keeps track of all process and thread information in a set of data structures
- New process creation steps
  - Upon creation of a process, a kernel process object is created
  - An Executive process object is created
  - The thread that created the process is given a handle to the executive object
LSN 6 – Creating Processes in Windows

- Win32 API for creating processes

```c
CreateProcess( LPCTSTR lpApplicationName,
               LPSTR lpCommandLine,
               LPSECURITY_ATTRIBUTES lpProcessAttributes,
               LPSECURITY_ATTRIBUTES lpThreadAttributes,
               BOOL bInheritHandles,
               DWORD dwCreationFlags,
               LPVOID lpEnvironment,
               LPCTSTR lpCurrentDirectory,
               LPSTARTUPINFO lpStartupInfo,
               LPROCESS_INFORMATION lpProcessInformation
           );
```
• Win32 API for creating threads

    CreateThread( LPSECURITY_ATTRIBUTES lpThreadAttributes,
                 DWORD dwStackSize,
                 LPTHREAD_START_ROUTINE lpStartAddress,
                 LPVOID lpParameter,
                 DWORD dwCreationFlags,
                 LPDWORD lpThreadId );
LSN 6 – Homework

• Assignment – HW 3
  – Complete HMWRK3.pdf

• References
  – Help within MSVS for specific function syntax descriptions
  – http://www.devarticles.com/c/a/Cplusplus/Multithreading-in-C/