WinNT Structure

- Kernel:
  - Executives
  - Win32 GDI
- User level:
  - System processes
    - services.exe, winlogon.exe, smss.exe, csrss.exe
  - Services
  - Environmental subsystems
- NTDLL.DLL

Objects

- an object is a single, runtime instance of a statically designed object type
  - Examples: process, thread, section, file, port, access token, event, event pair, semaphore, mutant, timer, obj dir, symlink, profile, key

Object Handles

- ways to get handles:
  - create the object
  - ask kernel for a handle
  - inherit the handle from parent process
  - being passed a duplicated handle
- handles are pointers to process object table
  - object table entry: granted access, inheritance flag, pointer to the object
    - creator of an object decides whether its handle can be inherited

Object Protection

- Security description:
  - Owner SID
  - Group SID
  - Discretionary ACL
  - System access control list: what should be logged
- each authenticated user has an access token, attached to all his or her threads
- user access token is only checked at handle granting time; not at every access

Processes and Threads

- Processes and threads are implemented as objects
- provide multitasking and multi-threading capabilities
- process/thread objects contain exception ports/termination port
  - for sending messages to other processes, e.g. environment subsystem
- thread can be alerted

Synchronization Objects

- Process, threads, file, event, event pair, semaphore, timer, mutant
- Signalled and non-signalled state
- Different types of objects have different rules for waking-up

LPC

- optimized RPC for client/server
  - sending a message to a port object
  - sending a message pointer to a port and passing the message in shared memory
  - passing a message to a particular server thread through a dedicated shared memory region
- “callbacks”

Virtual Memory

- Application interface: reserve and commit
  - aligned at 64KB boundary, multiple of 4KB
- Section object
  - support shared-memory and mapped files
  - section objects have ACLs
- Page protection modes
- Support copy-on-write
- Kernel space: paged pool and non-paged pool

Memory Manager Components

- Alloc/dealloc virtual memory
- page fault handler
- Kernel threads:
  - balance set manager
  - process/stack swapper
  - dirty page writer
  - dirty file page writer
  - dereference segment thread
  - zero-page thread

Physical Memory Management

- Paging file
- Working set limits (min/max) + FIFO + “Standby” list
- Page list dynamics

I/O System

- Support Asynchronous I/O
- IRP (I/O request packet)
  - Pass requests between file system and device drivers
- “Cache Manager”: cache file data
  - Use memory manager to cache file data
  - cache based on virtual addresses

NTFS

- Volume (partition)
- Clusters (blocks)
- Log all metadata changes
  - logfile
  - undo/redo