

# **CSE 451: Operating Systems**

## **Hard Lessons Learned**

### **Windows**

### **Objects, Handles and Ref Counts**

**Gary Kimura**

# Object Manager

- Windows OS was written in C. C doesn't have a nice way to deal with "Objects".
- So, we wrote an Object Manager where each component of the OS was responsible for defining its Object types with functions for creation, deletion, etc.
- The Object Manager also took care of the naming of objects.
- We had objects for processes, threads, memory, files, devices, events, etc.
- If a user created an object, it was given a handle to the object.

# Handles and References

OpenFile()

OpenFile()

Write()

CloseFile()

CloseFile()

HandleCount:

RefCount:

File Object



# Counts

- Each object had two counters, A **HandleCount** corresponding to how the user accesses the object, and a **RefCount** corresponding to when code in the kernel wants to access the object.
- Zero HandleCount means the user has “Closed” the object.
- Zero RefCount means the Kernel is done with the object and it can be deleted.
- Notion of delayed object deletion. A file might stay “open” for a while longer.

# Cleanup and Close

CloseFile()

HandleCount:

RefCount:



NtfsCleanup()

..DeferenceObject()

NtfsClose()

# Moral of the Story

- Who is to blame for the confusing names?
- Even the most “complete” Design documents will probably have gaps.
- Choose being doing a clean redesign or applying a patch.
- Designs need to allow for corrections, growth and flexibility.