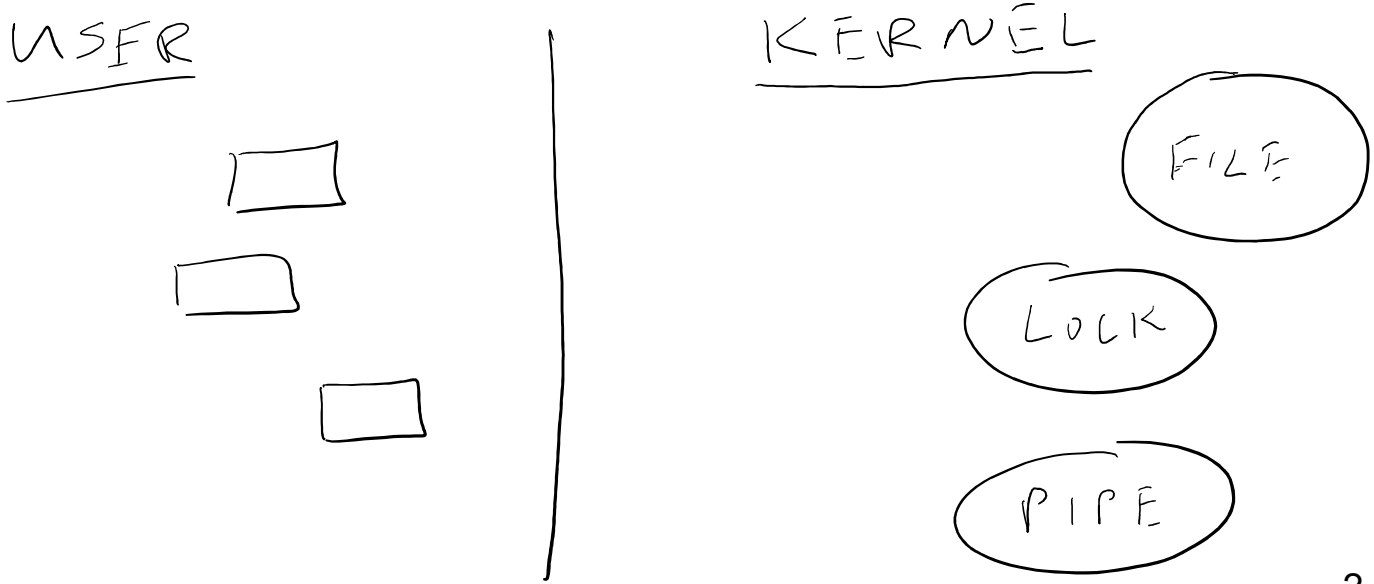
CSE 451: Operating Systems
Hard Lessons Learned

Windows
Handle Table

Gary Kimura

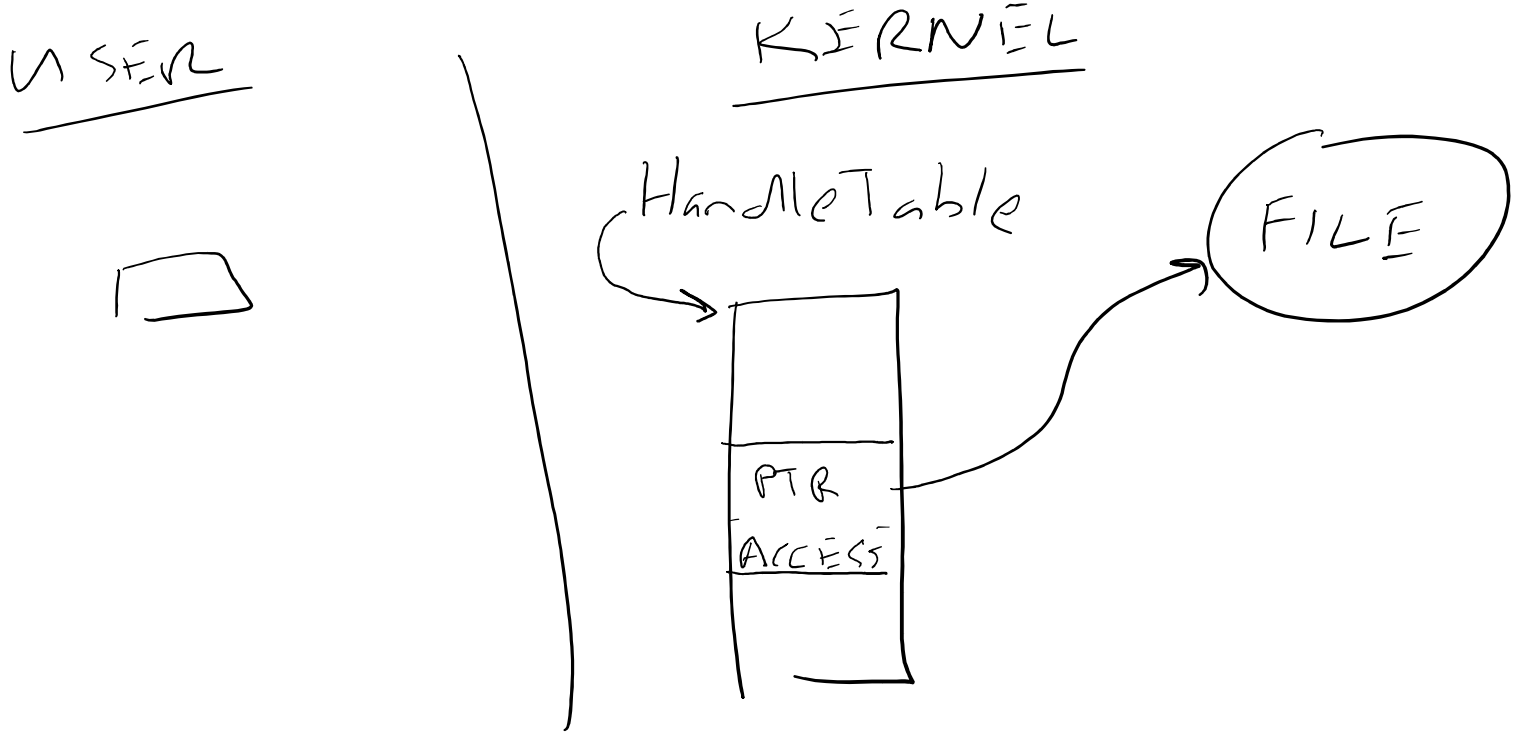
What is a handle?

- Users refer to kernel objects (those that it are allowed to see) using a handle provided by the OS
- For example, stdin, stdout, and stderr are considered handles
- In Windows there are handles for many things (files, processes, threads, events, locks, etc.)



What is a handle table?

- In Windows every process is given a process that is maintained by the OS.
- Where the handle value is used by the kernel as an index to this table.



What is a good table size?

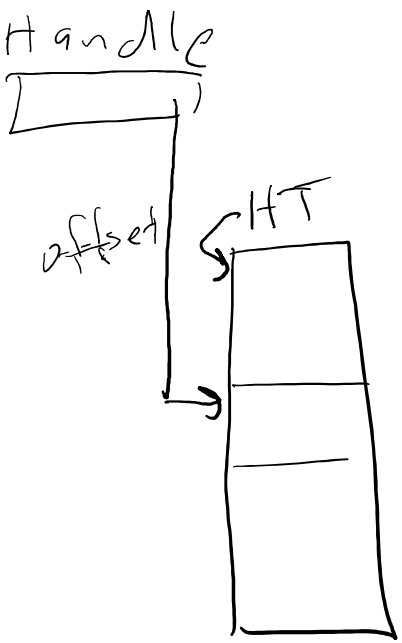
- For typically processes less than 100 entries should suffice.
- However, some apps use a lot of handles. And there are benchmarks that test the speed of creating and deleting hundreds of thousands of handles.

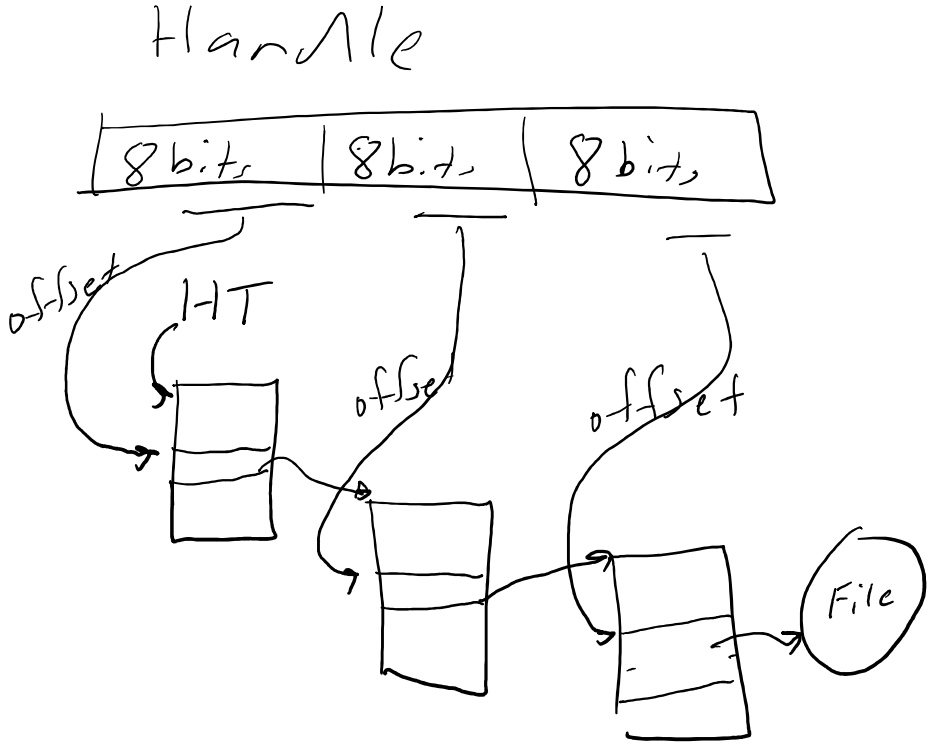
So what did Windows do?

- Windows started allocating a linear array for each processes handle table (think malloc()).
- And if it needed to grow the OS would realloc() the new size. The code would double the size each time it needed to grow. (think 32, 64, 128, 256, 512, …)
- Sounds simple, except when the size gets really large.
- This presented two problems
  - Finding enough contiguous virtual address space to grow the table
  - Having to copy over the old table on to the newly allocated table was time consuming
- Consequently Windows was very slow growing at this

How did it get fixed?

- By using a multi-level table instead of a single level table. This is similar to a multi-level page table.





Did the fix work?

- It vastly improved how Windows performs