## Mapped Files

**Mapped files** enable processes to do file I/O using loads and stores Instead of "open, read into buffer, operate on buffer, ..."

- → Bind a file to a virtual memory region (see Unix mmap)
  - PTEs map virtual addresses to physical frames holding file data
  - Virtual address base + N refers to offset N in file

Initially, all pages mapped to file are invalid (similar to a swapped page)

- OS reads a page from file when invalid page is accessed
- · OS writes a page to file when evicted, or region unmapped
- If page is not dirty (has not been written to), no write needed (another use of the dirty bit in PTE)

## Advantages and drawbacks of mapped files

- → File is essentially backing store for that region of the virtual address space (instead of using the swap file)
- ✓ Uniform access for files and memory (just use pointers)
- Process has less control over data movement
  OS handles faults transparently
- Does not generalize to streamed I/O (pipes, sockets, etc.)