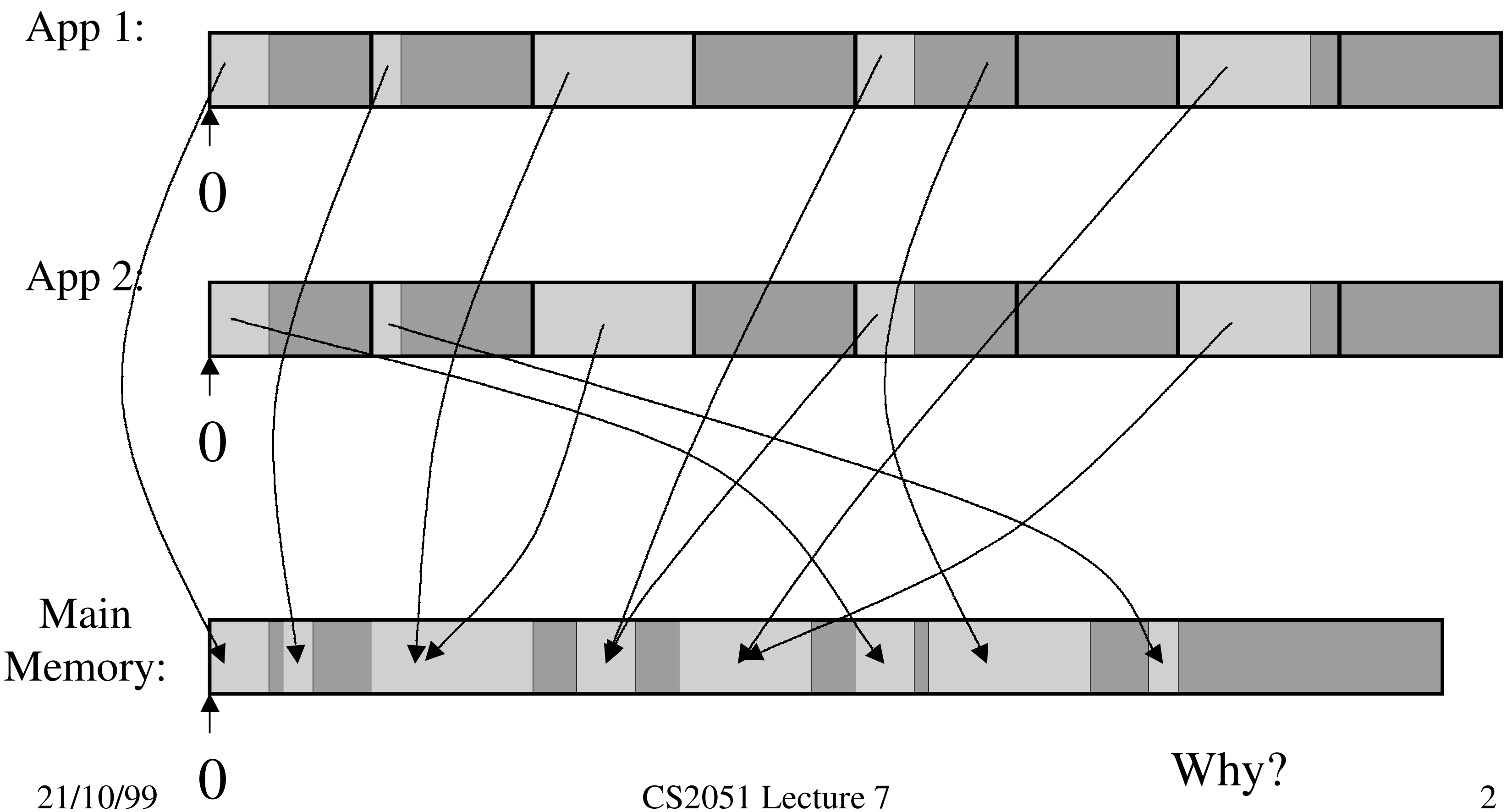
Efficient Use of Main Memory

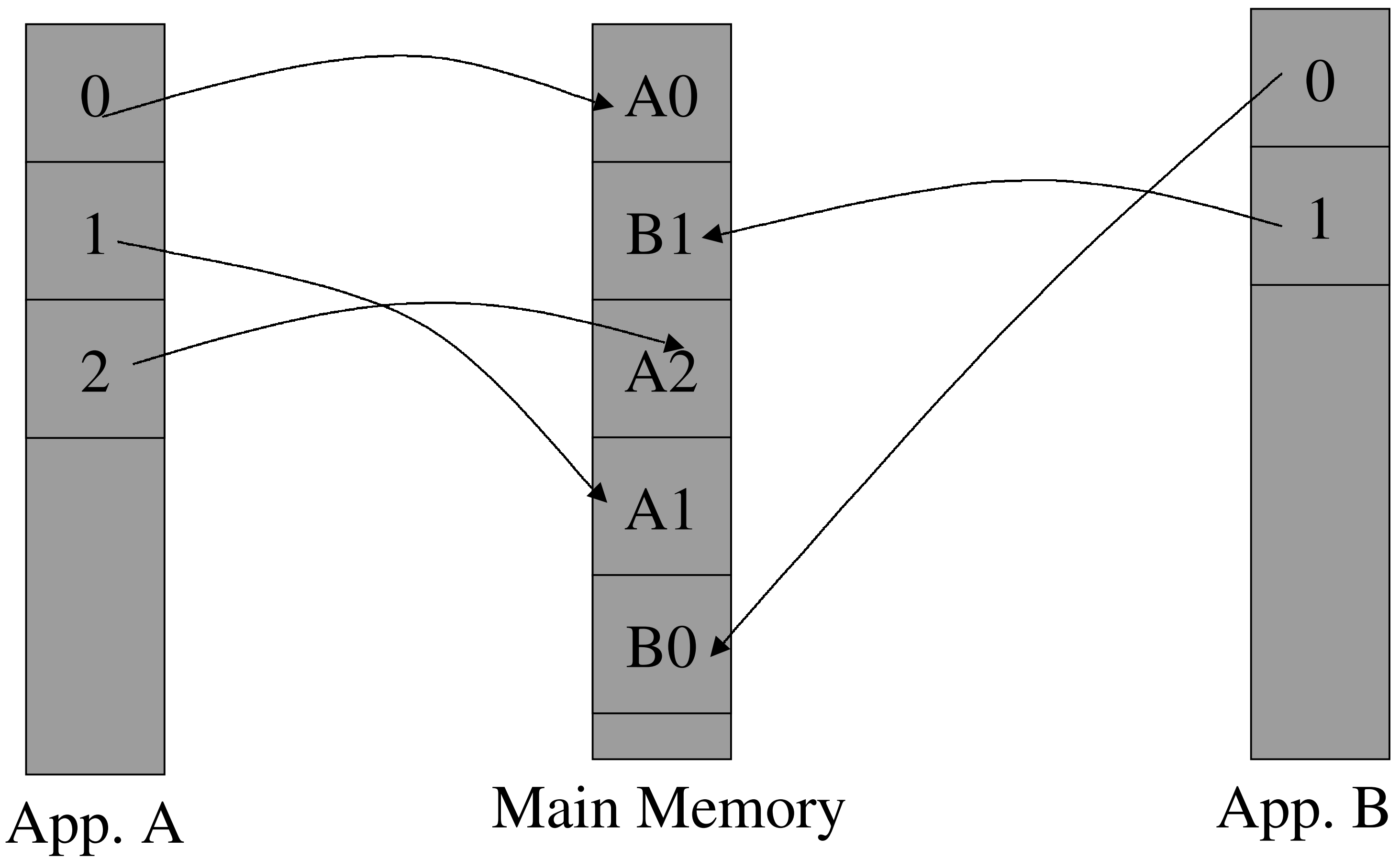
Towards Modern Operating Systems

Segmentation: Inefficient use of main memory

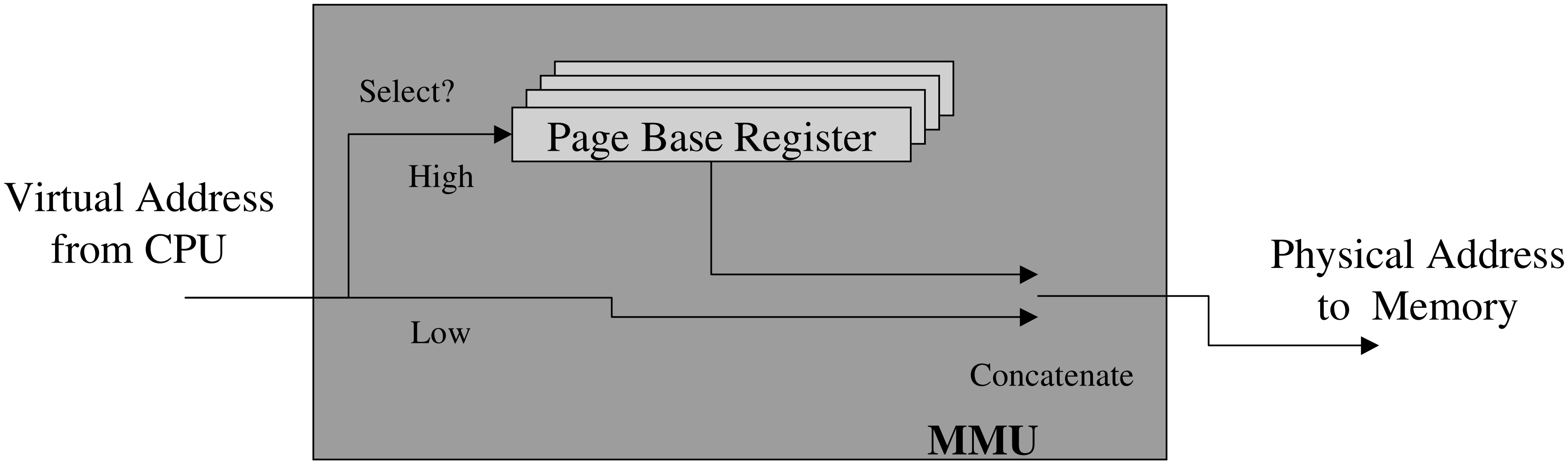


Why?

Paging - Fixed Sized Blocks



An MMU for Paging



Note:

– Concatenation, not addition

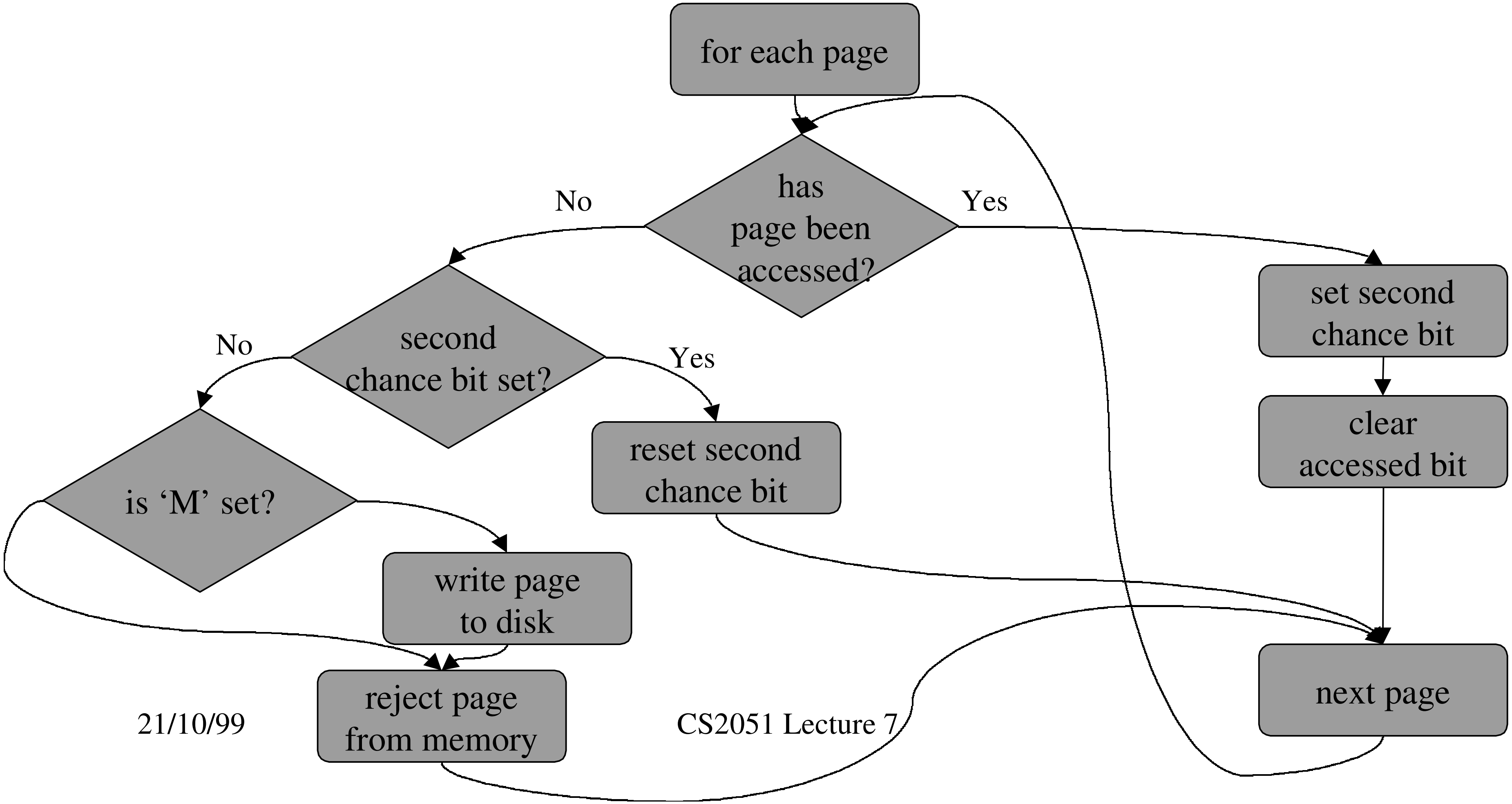
Virtual Memory Concept

- Not all application memory needed at once
- MMU implements
  - Presence bit, generates exception (when?)
  - Modified bit, set by MMU (when?)
  - Accessed bit, set by MMU (when?)
- Disk, Main Memory shared transparently by OS
- Page is fixed sized chunk (512 bytes to 8K bytes)

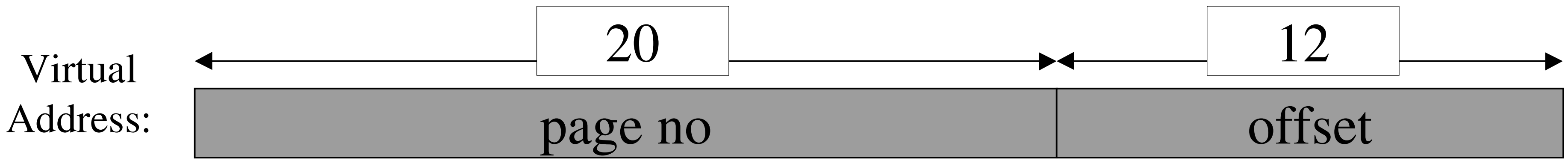
Page Rejection

- Wish to reject page that is unneeded for longest future time
  - Predict Future?
- Instead, use *Working Set* concept
- Reject pages that have Not been Used Recently (NRU algorithm)
- (LRU is fiendishly difficult!)

Not Recently Used with Second Chance algorithm



Page Tables may be large



- Page size is 4K bytes
- 2^{20} (1M) pages per process
- 4 bytes per page table entry
- 4 M bytes of page table *per process!*