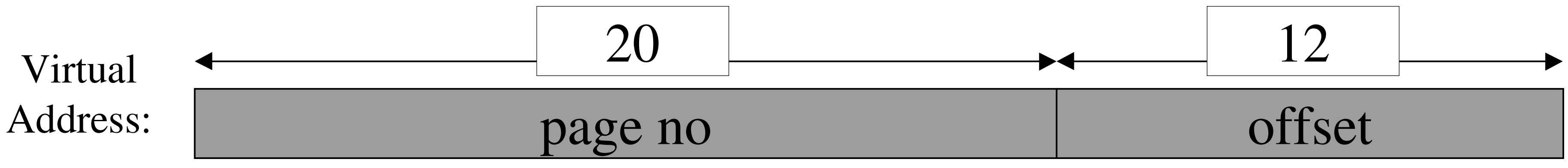
Page Tables may be large

Example numbers for 4K
page size, 32-bit addresses



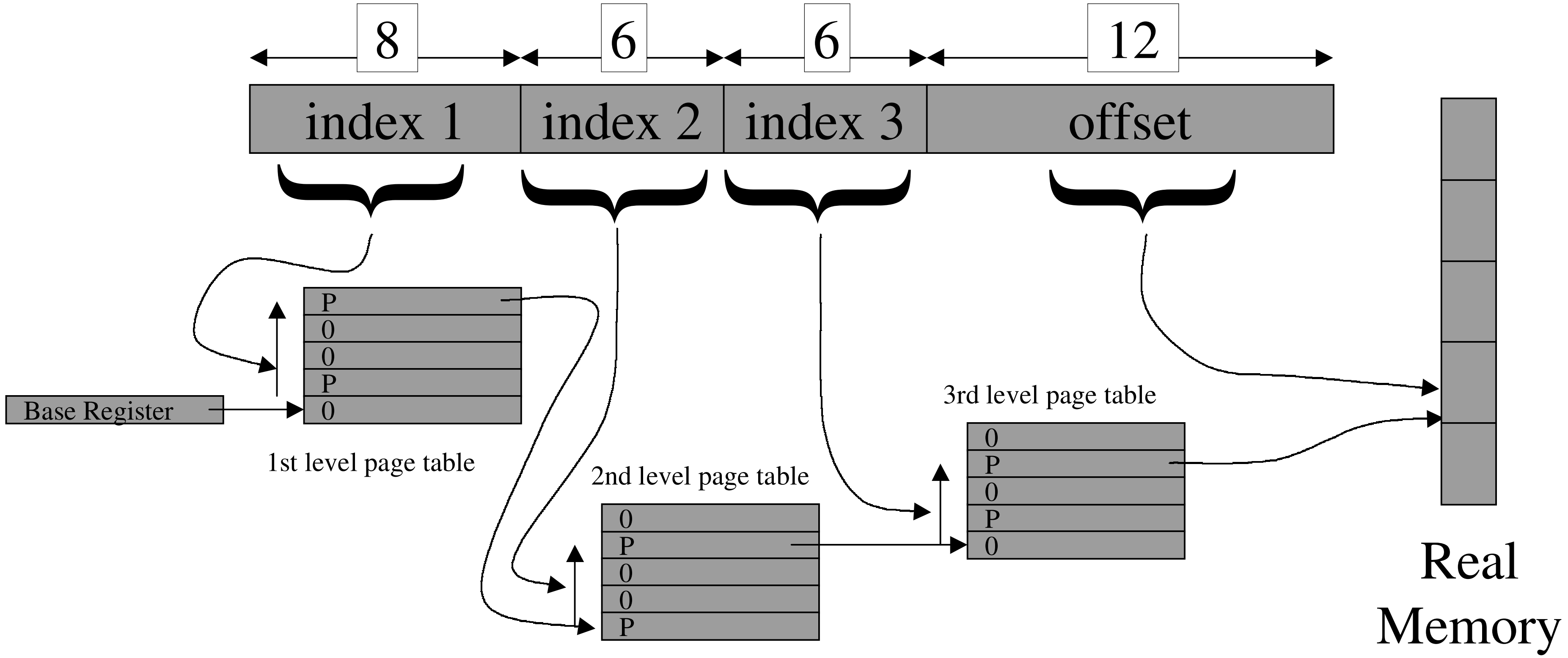
- Page size is 4K bytes
- 2^{20} (1M) pages per process
- 4 bytes per page table entry
- 4 M bytes of page table *per process!*

Insight:

- Most processes are smaller than 4 Gbyte
- Most page table entries are zero
  - and never accessed
- Don't need them in main memory
- Therefore: page the page tables!

Paged Page Tables: e.g. SPARC V8 Reference MMU

Virtual address:

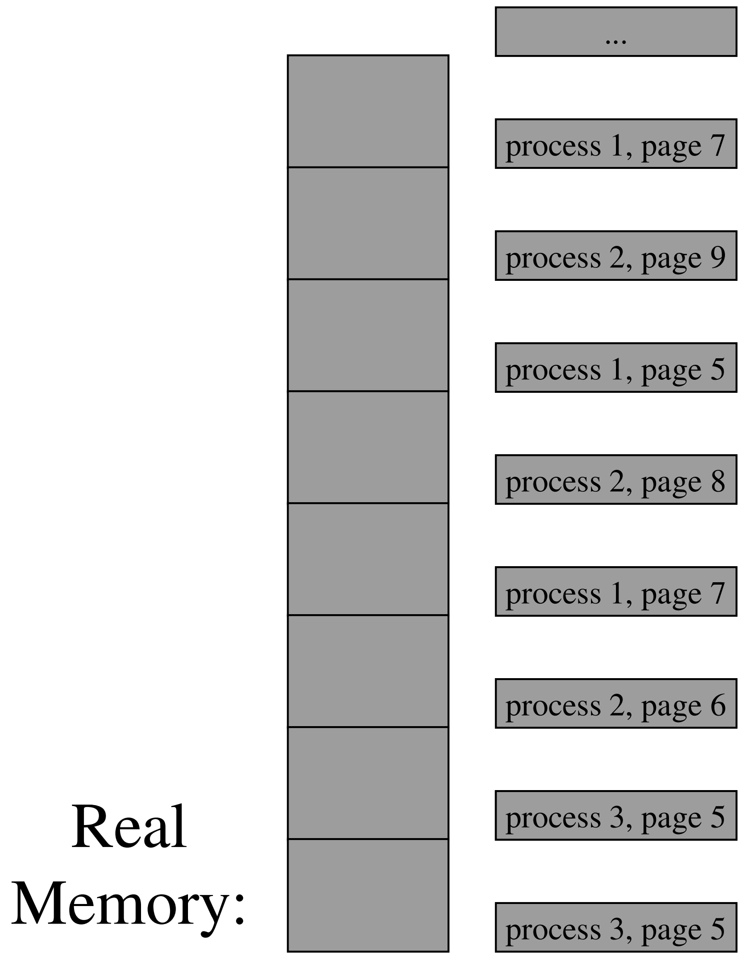


Minimizing MMU size

- Keep subset of page table entries in MMU
- Search all in parallel
- Miss in MMU may:
  - load another entry from page table in memory
  - page fault
- Table walk in:
  - Hardware (e.g. Sun), or Software (e.g. MIPS)
- See CS2042 for detail

Inverted Page Table

- OS probably needs extra data structures for disk-based pages
- MMU only needs to know about pages in main memory
- e.g. HP Precision

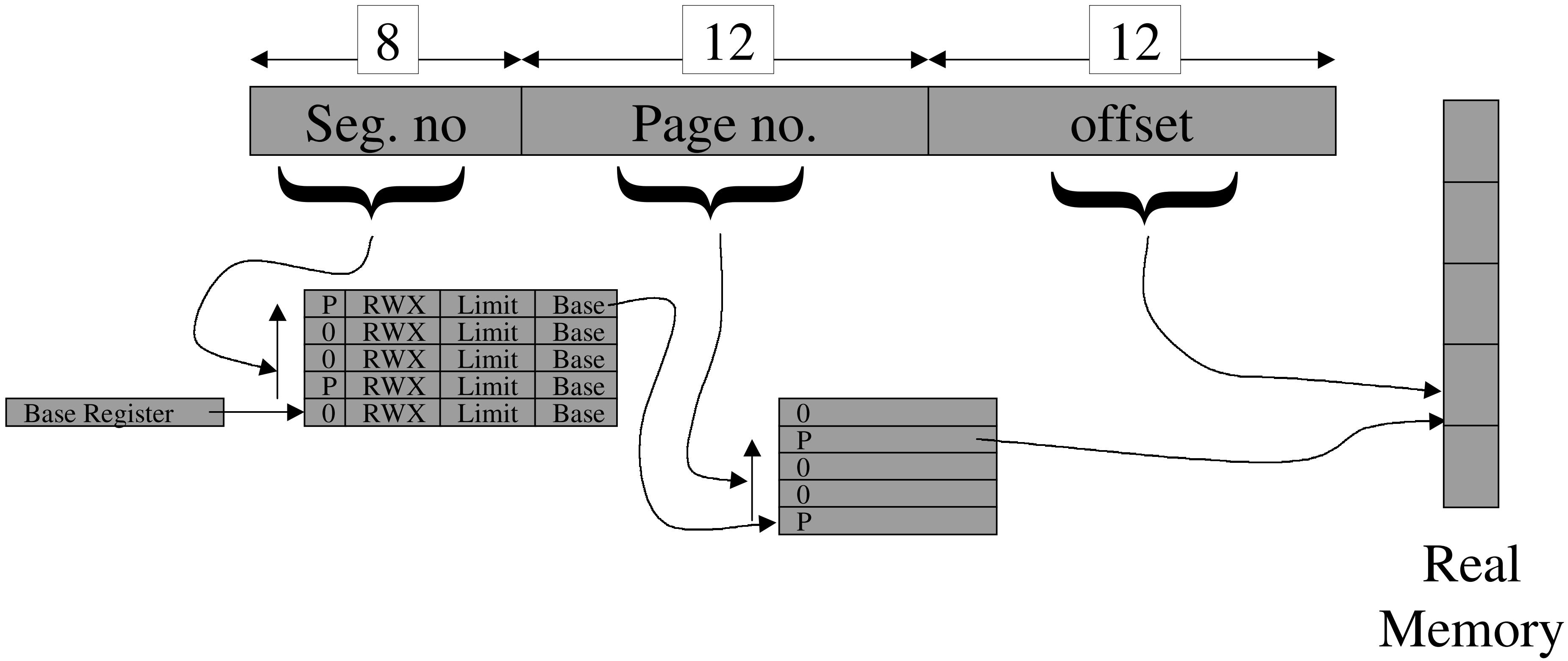


Summary

- Segmentation
  - Logical part of address space
  - Variable size (large for convenience)
  - Natural unit for sharing, protection
  - supports Virtual Memory
- Paging
  - One or many to form logical part
  - Fixed size for efficiency
  - unnatural unit
  - supports Virtual Memory

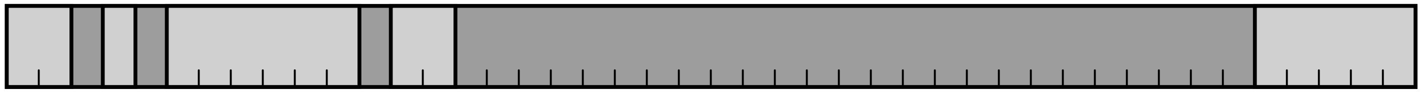
Hardware segments and paging

Virtual address:



Hardware Paging, Software Segments

Virtual address space



- Page table entries need RWX bits for sharing
- Every *segment* is whole number of pages
- Note *guard* pages