Distributed Operating Systems

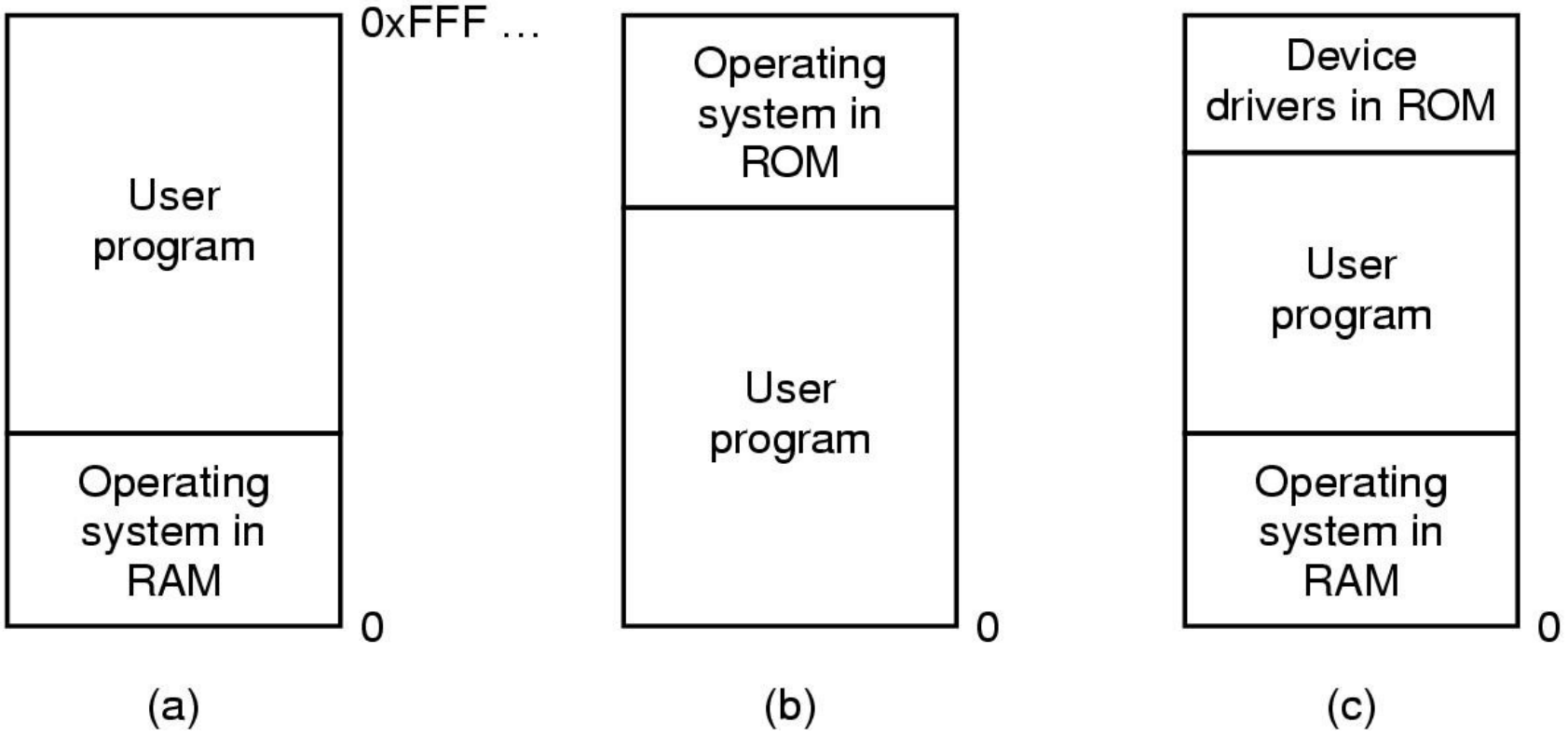
Memory Management

Topics

- Basic Memory Management
- Relocation and Protection
- Swapping
- Paging
- Page Replacement Algorithms

Basic Memory Management

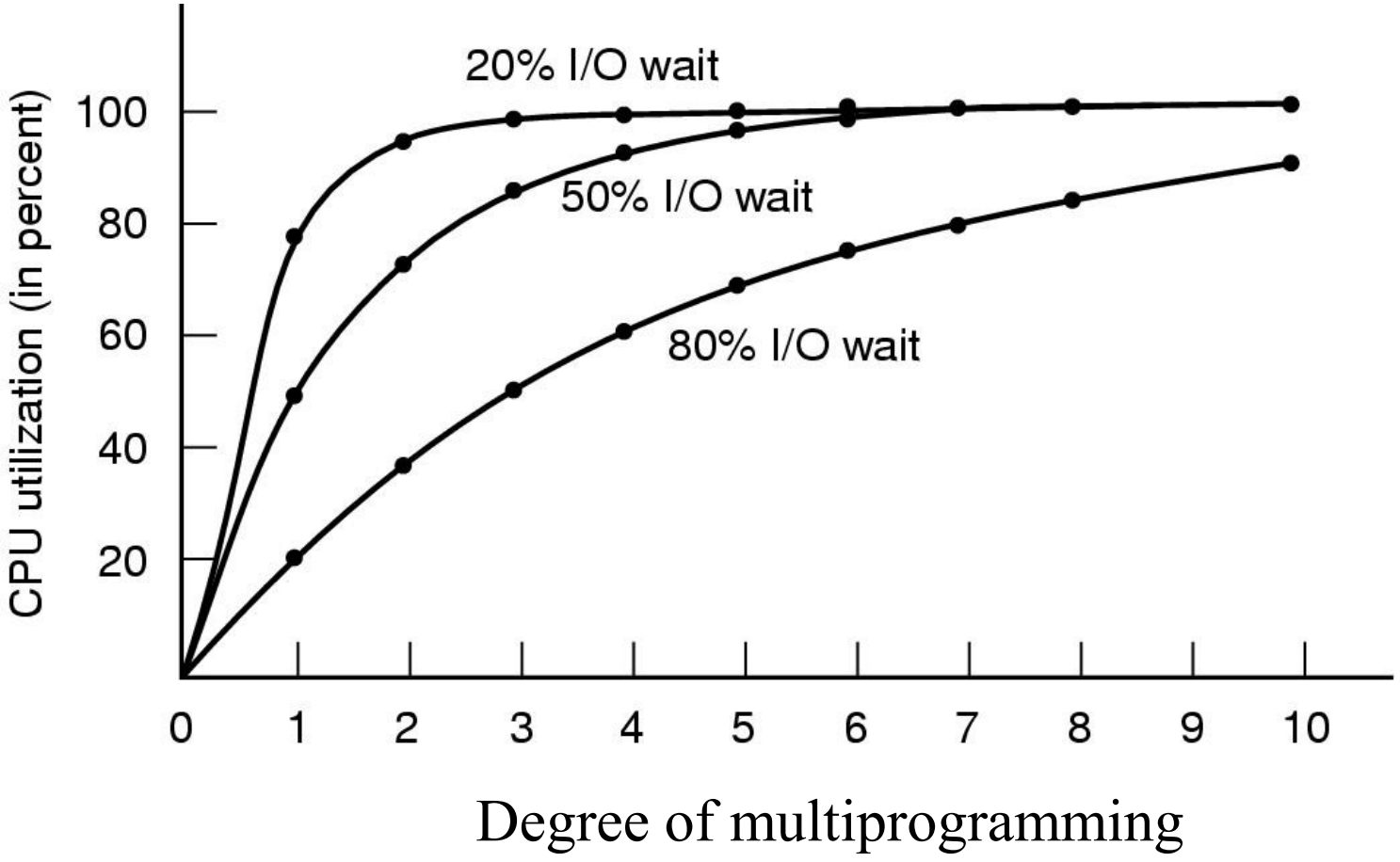
Monoprogramming without Swapping or Paging



Three simple ways of organizing memory

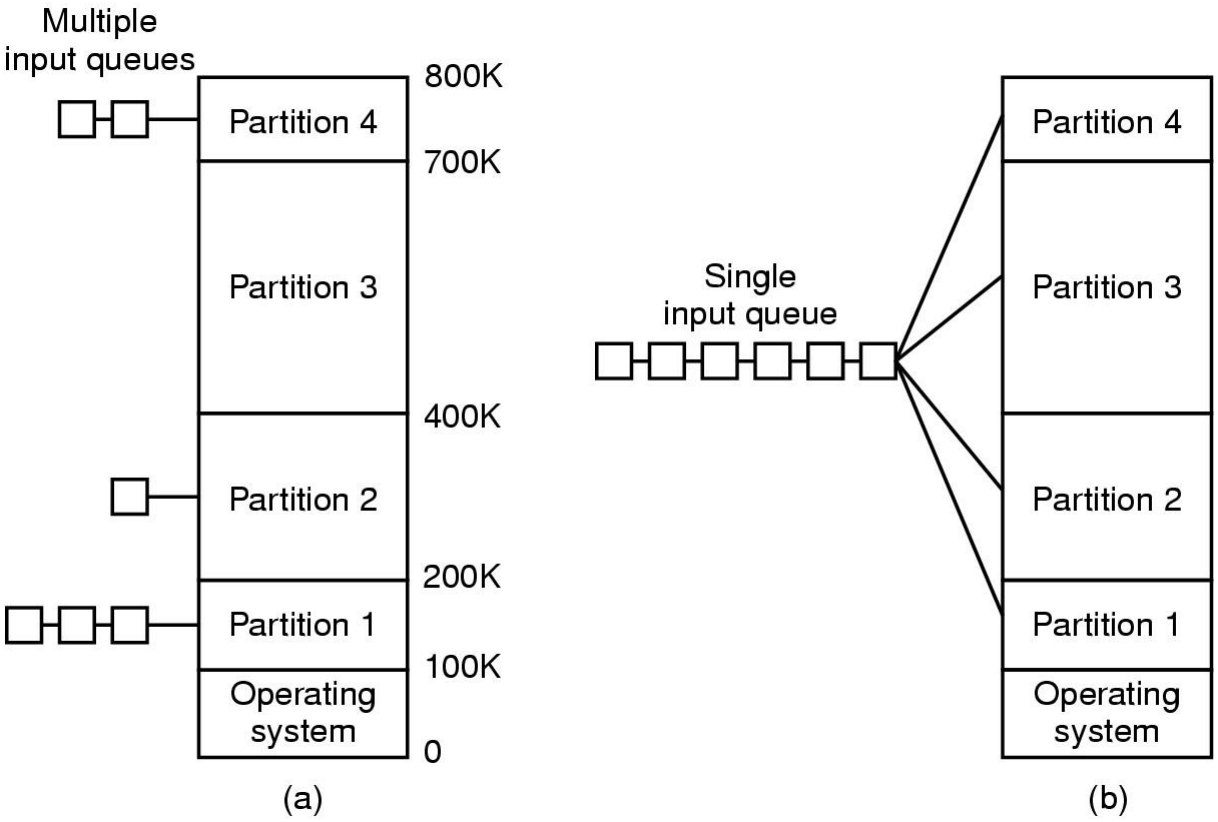
- an operating system with one user process

Modeling Multiprogramming



CPU utilization as a function of number of processes in memory

Multiprogramming with Fixed Partitions



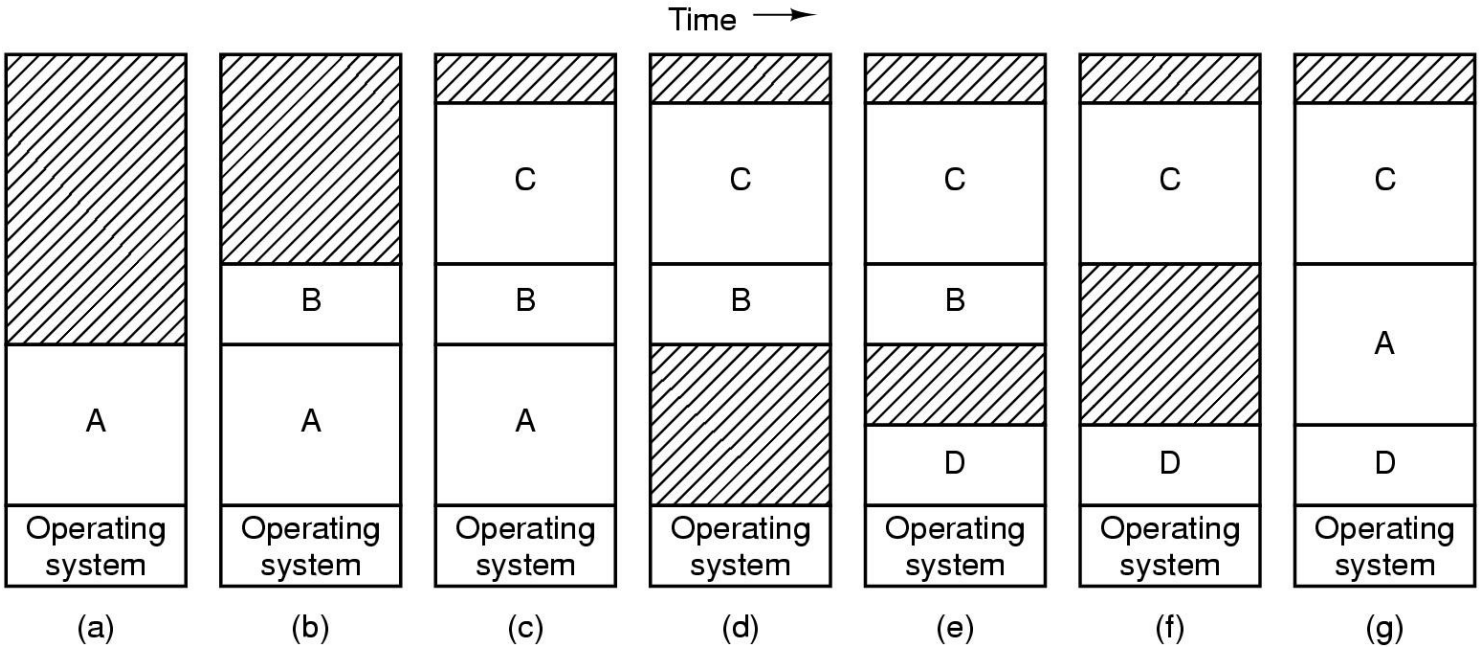
- Fixed memory partitions
  - separate input queues for each partition
  - single input queue

Relocation and Protection

- Cannot be sure where program will be loaded in memory
  - address locations of variables, code routines cannot be absolute
  - must keep a program out of other processes' partitions

- Use base and limit values
  - address locations added to base value to map to physical addr
  - address locations larger than limit value is an error

Swapping (1)

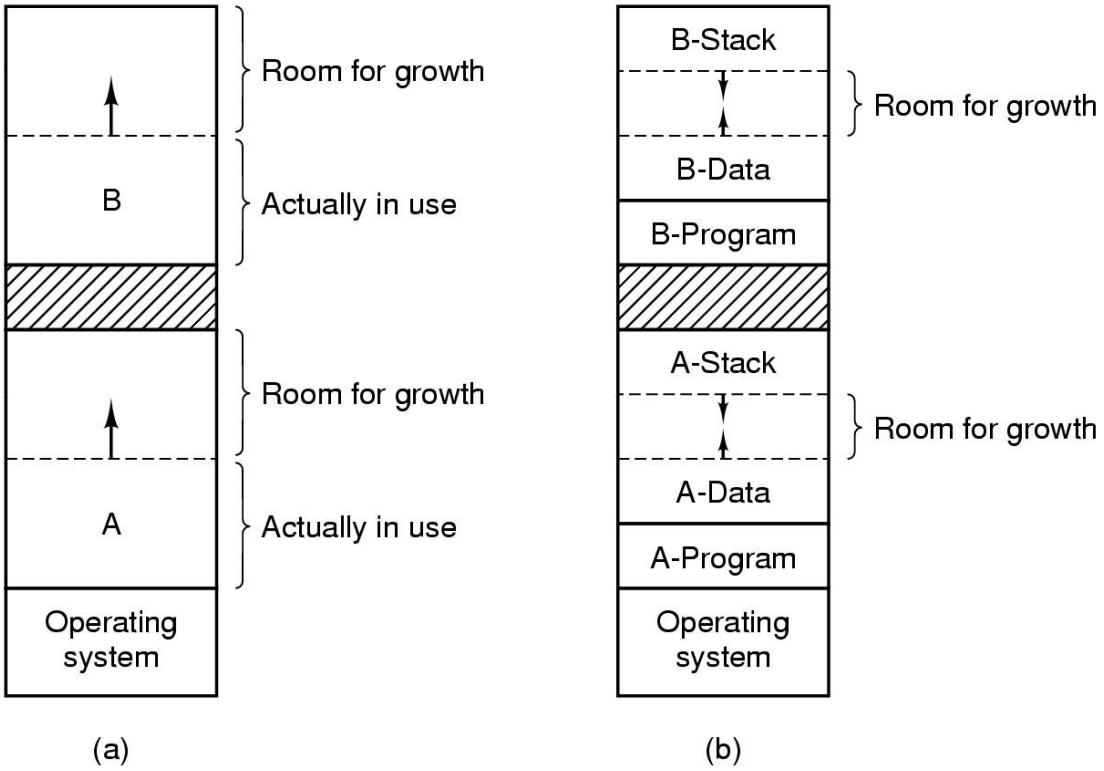


Memory allocation changes as

- processes come into memory
- leave memory

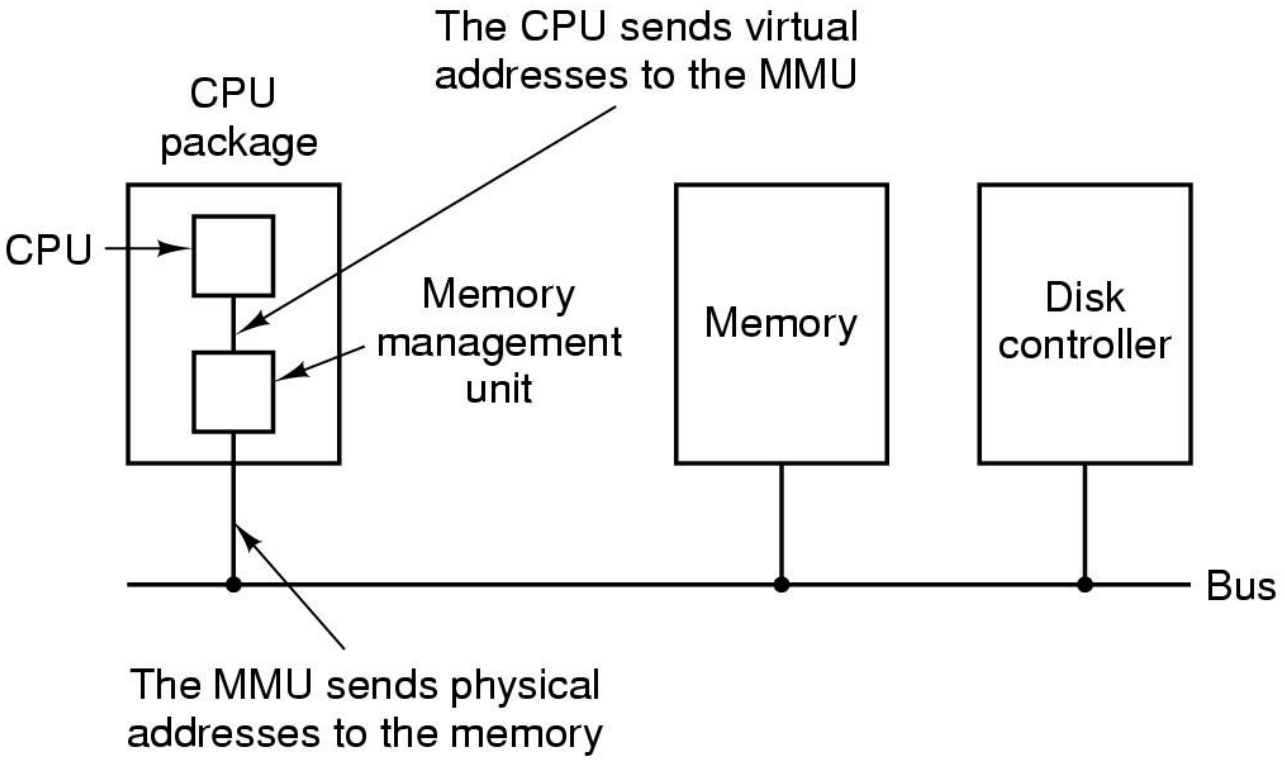
Shaded regions are unused memory

Swapping (2)



- Allocating space for growing data segment
- Allocating space for growing stack & data segment

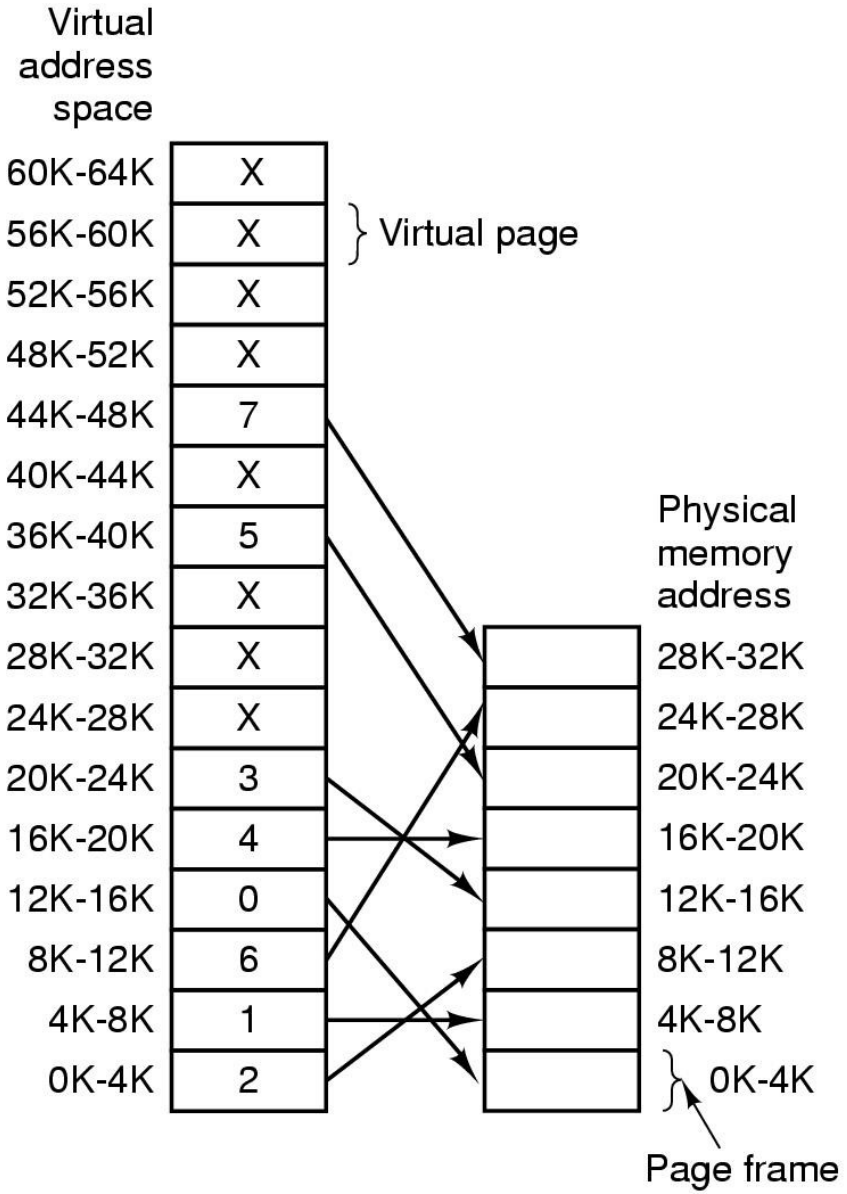
Virtual Memory
Paging (1)



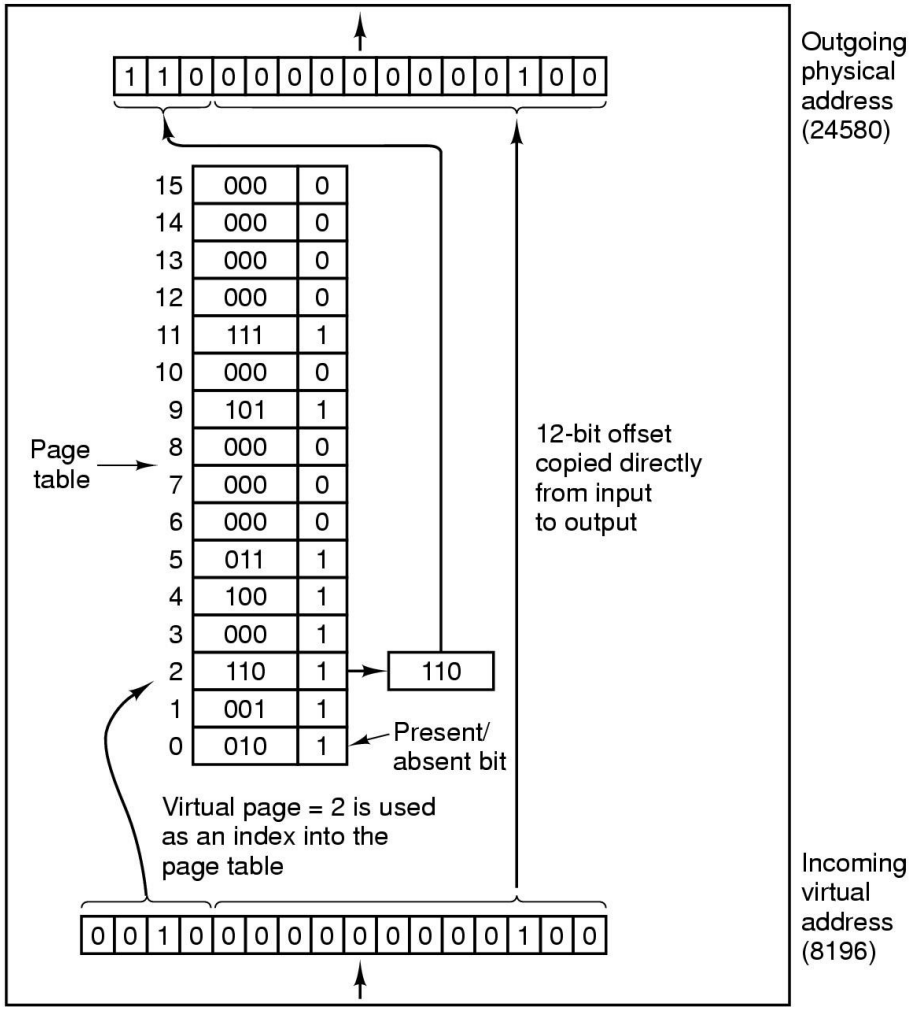
The position and function of the MMU

Paging (2)

The relation between virtual addresses and physical memory addresses are given by page table

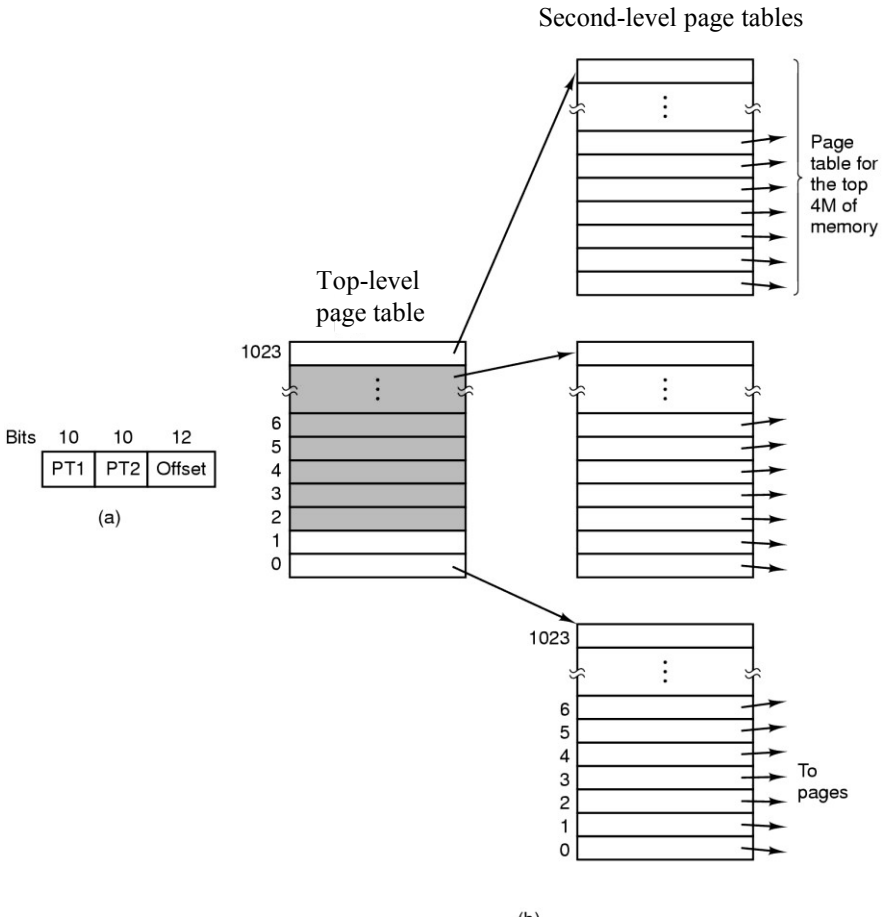


Page Tables (1)



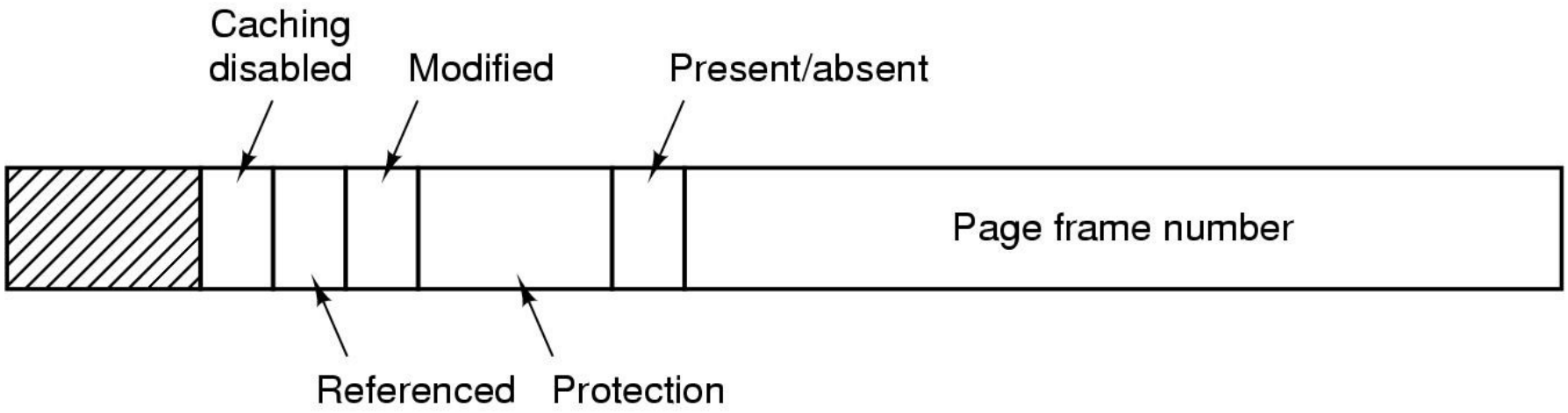
Internal operation of MMU with 16, 4 KB pages

Page Tables (2)



- 32 bit address with 2 page table fields
- Two-level page tables

Page Tables (3)



Typical page table entry

TLBs – Translation Lookaside Buffers

Valid	Virtual page	Modified	Protection	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

A TLB to speed up paging

Page Fault Handling

1. Hardware traps to kernel
2. General registers saved
3. OS determines which virtual page needed
4. OS checks validity of address, seeks page frame
5. If selected frame is dirty, write it to disk

Page Replacement Algorithms

- Page fault forces choice
  - which page must be removed
  - make room for incoming page
- Modified page must first be saved
  - unmodified just overwritten
- Better not to choose an often used page
  - will probably need to be brought back in soon

Optimal Page Replacement Algorithm

- Replace page needed at the farthest point in future
  - Optimal but unrealizable

- Estimate by …
  - logging page use on previous runs of process
  - although this is impractical

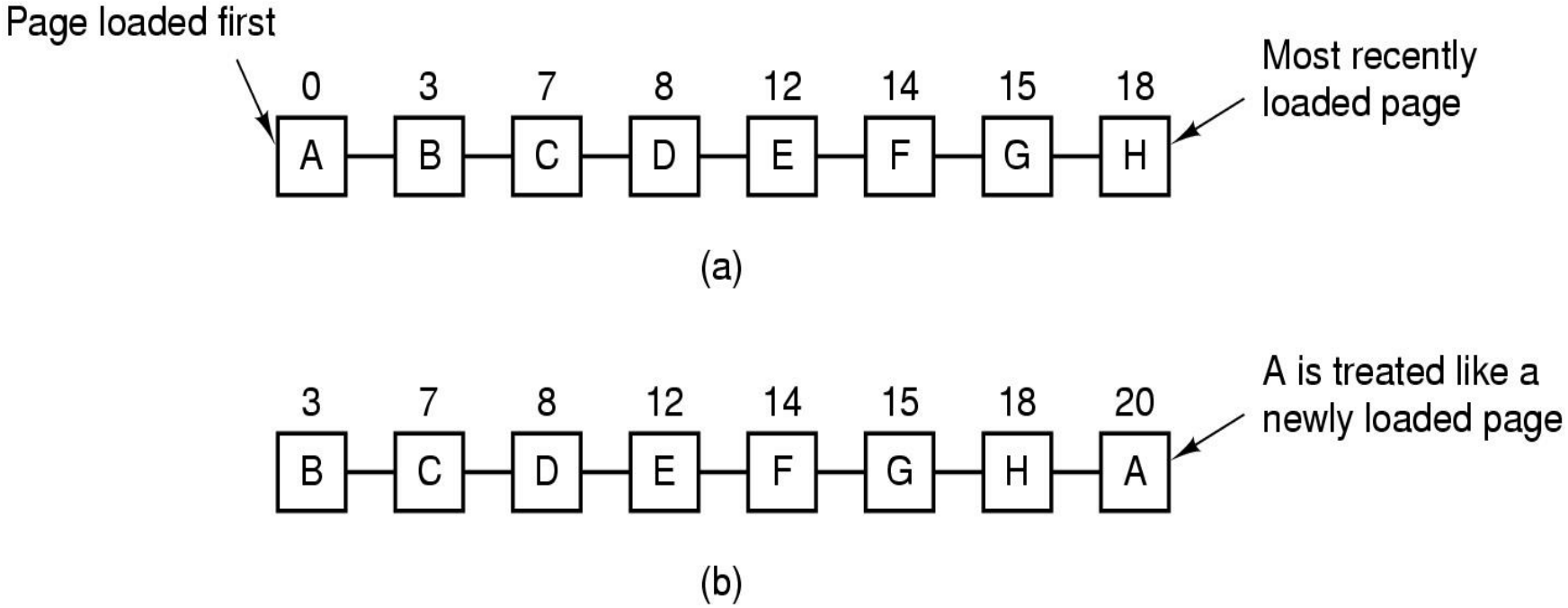
Not Recently Used Page Replacement Algorithm

- Each page has Reference bit, Modified bit
  - bits are set when page is referenced, modified
- Pages are classified
  1. not referenced, not modified
  2. not referenced, modified
  3. referenced, not modified
  4. referenced, modified
- NRU removes page at random
  - from lowest numbered non empty class

FIFO Page Replacement Algorithm

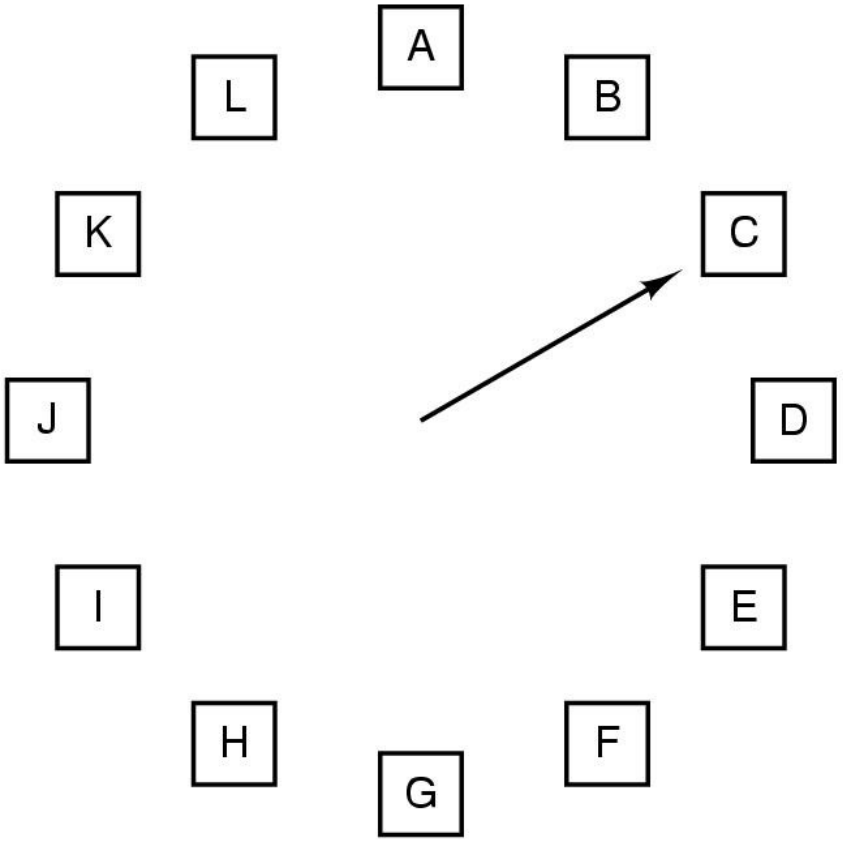
- Maintain a linked list of all pages
  - in order they came into memory
- Page at beginning of list replaced
- Disadvantage
  - page in memory the longest may be often used

Second Chance Page Replacement Algorithm



- Operation of a second chance
  - pages sorted in FIFO order
  - Page list if fault occurs at time 20, *A* has *R* bit set (numbers above pages are loading times)

The Clock Page Replacement Algorithm



When a page fault occurs, the page the hand is pointing to is inspected. The action taken depends on the R bit:

R = 0: Evict the page

R = 1: Clear R and advance hand

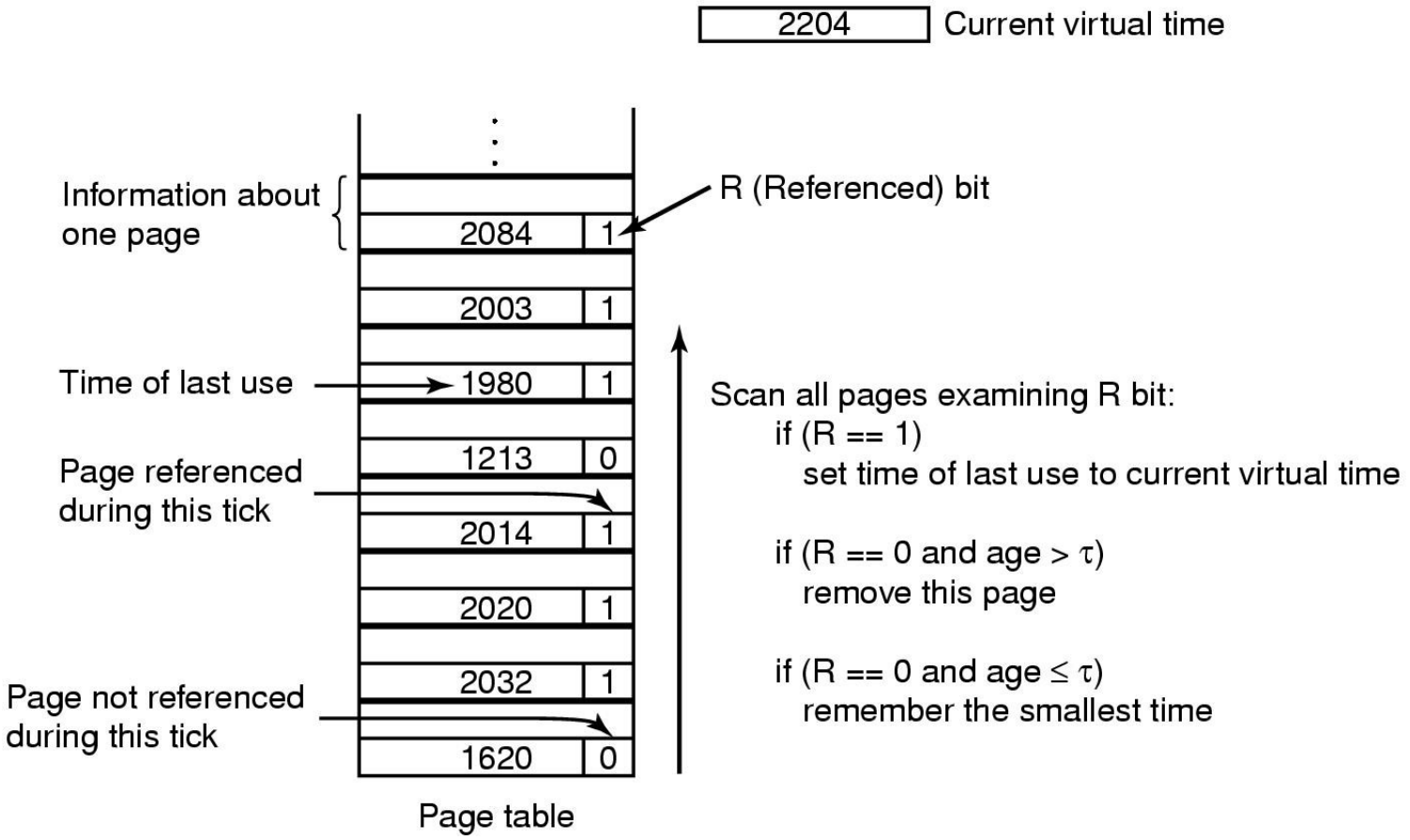
Least Recently Used (LRU)

- Assume pages used recently will be used again soon
  - throw out page that has been unused for longest time

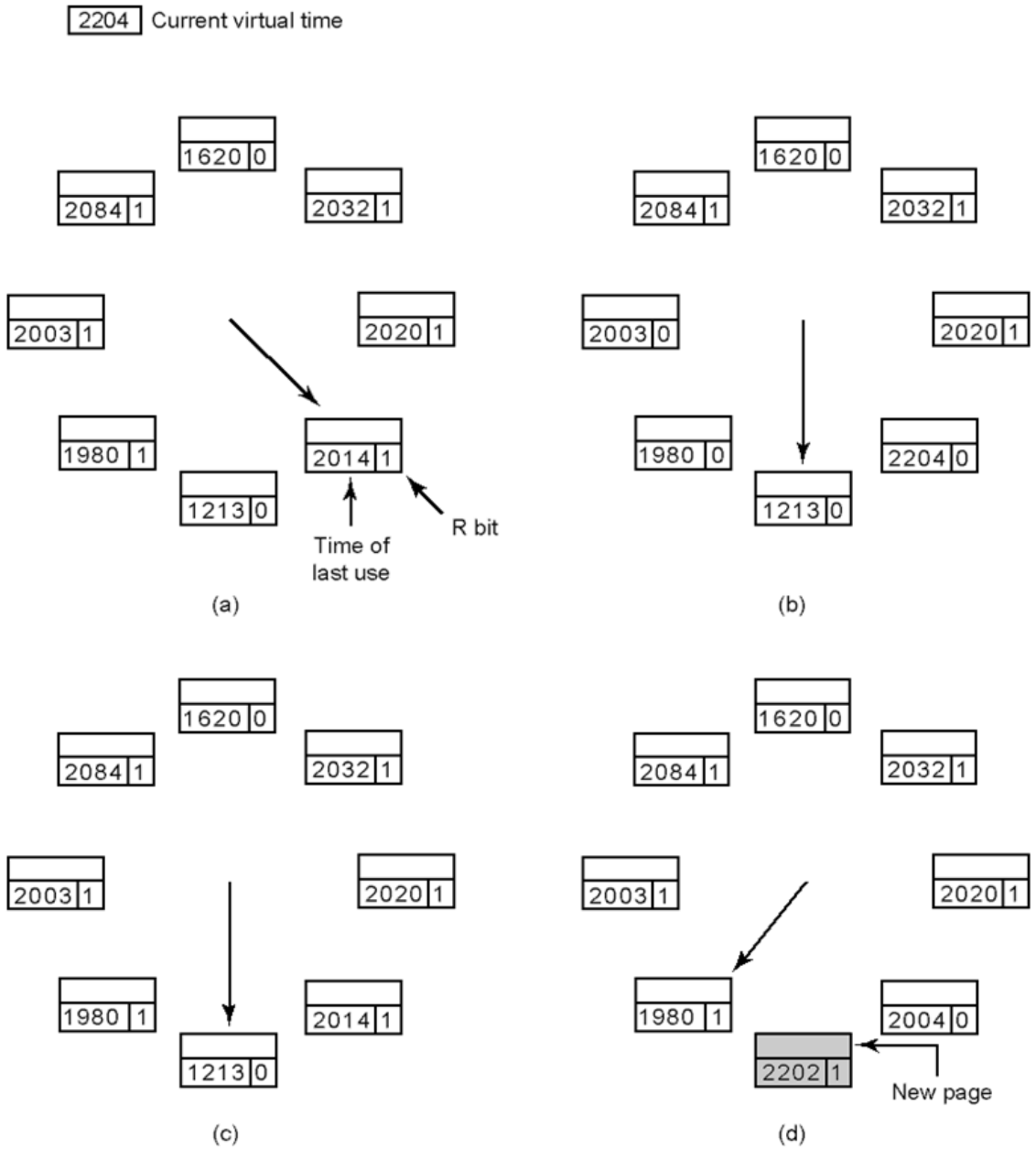
- Must keep a linked list of pages
  - most recently used at front, least at rear
  - update this list every memory reference !!

- Alternatively keep counter in each page table entry
  - choose page with lowest value counter
  - periodically zero the counter

The Working Set Page Replacement Algorithm



The working set algorithm



Operation of the WSClock algorithm

Questions?