STEVENS INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

CS 492: Operating Systems

Page Replacement (3)

Instructor: Iraklis Tsekourakis

Email: itsekour@stevens.edu

Recap

- Page Replacement
  - Page fault
  - Page replacement polices
    - Optimal
    - Least Recently Used (LRU)
    - FIFO
    - Not Recently Used (NRU)

Goal of This Part of Lecture

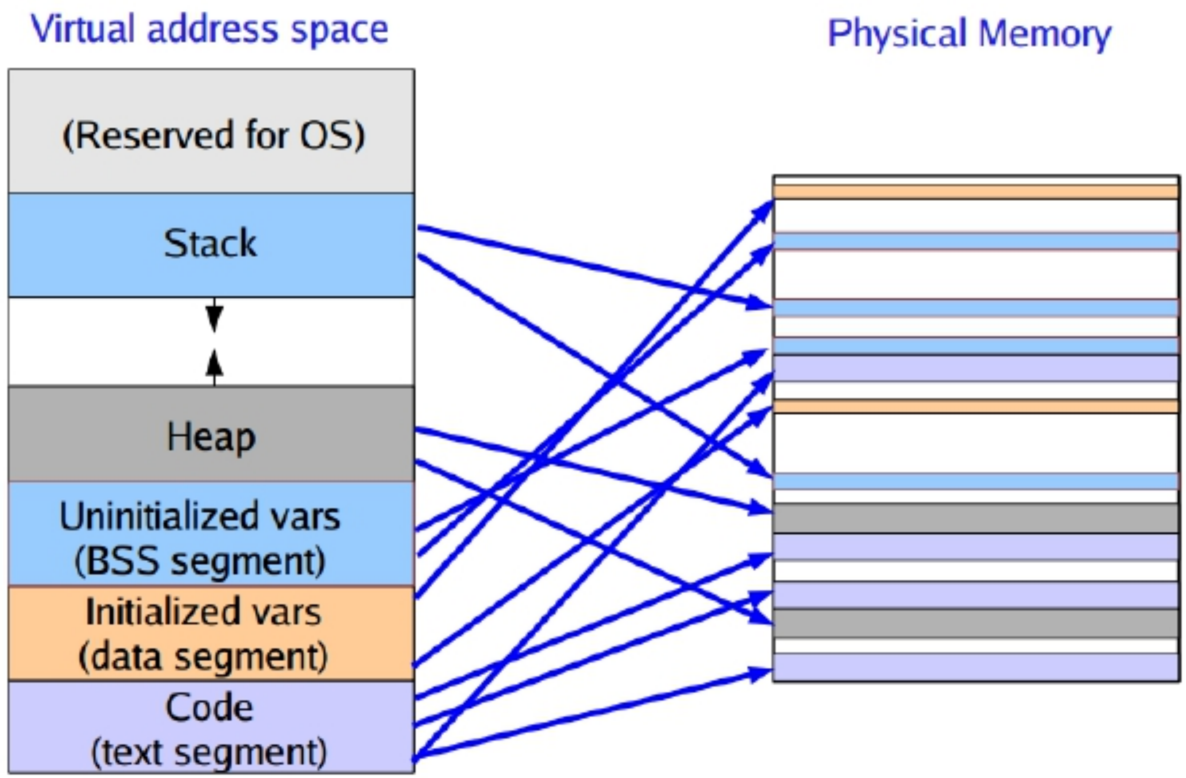
- Page replacement polices
  - Working set model

Fetch Policy 1: Demand Paging

- *Demand paging* – pages are loaded on demand, not in advance

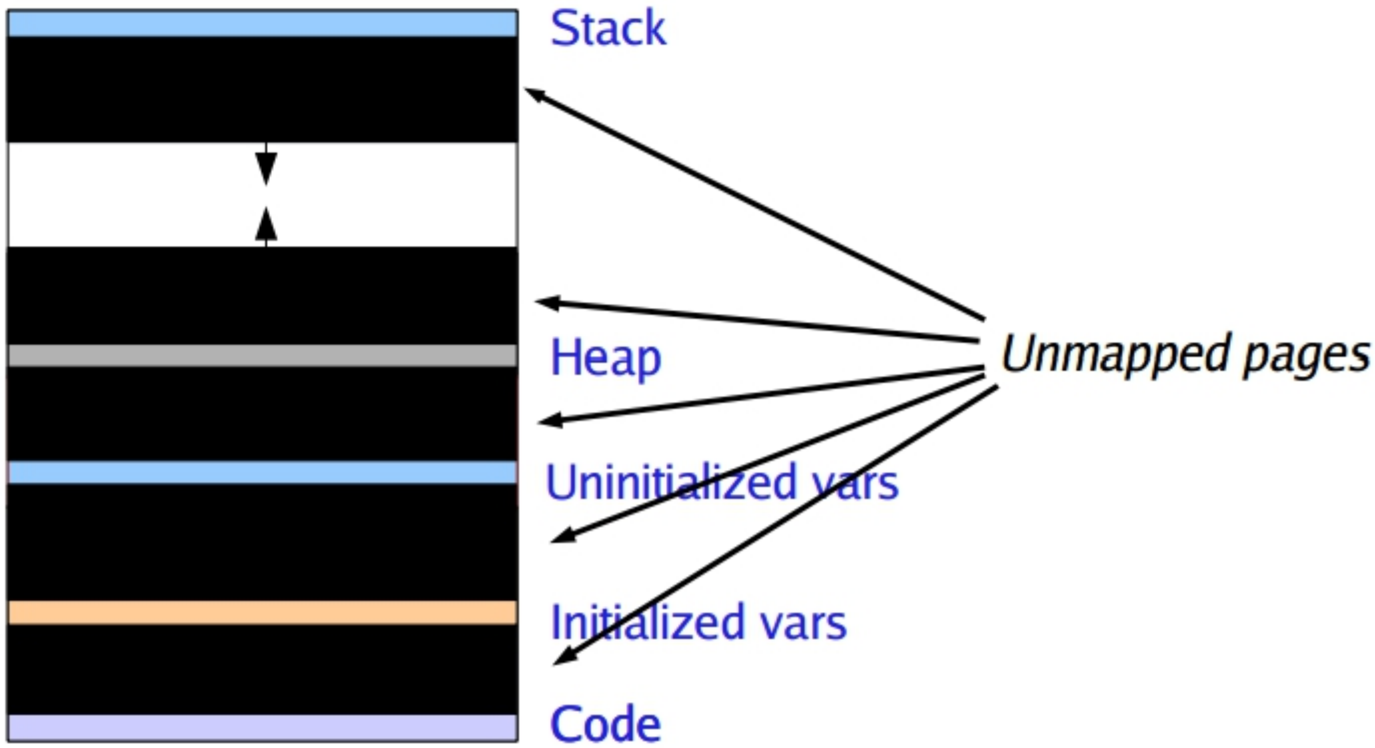
Demand Paging

- It does not make sense to read an entire program into memory at once
  - Only a small portion of a program's code may be used!
  - For example, if you never use the “save as PDF” feature in OpenOffice...



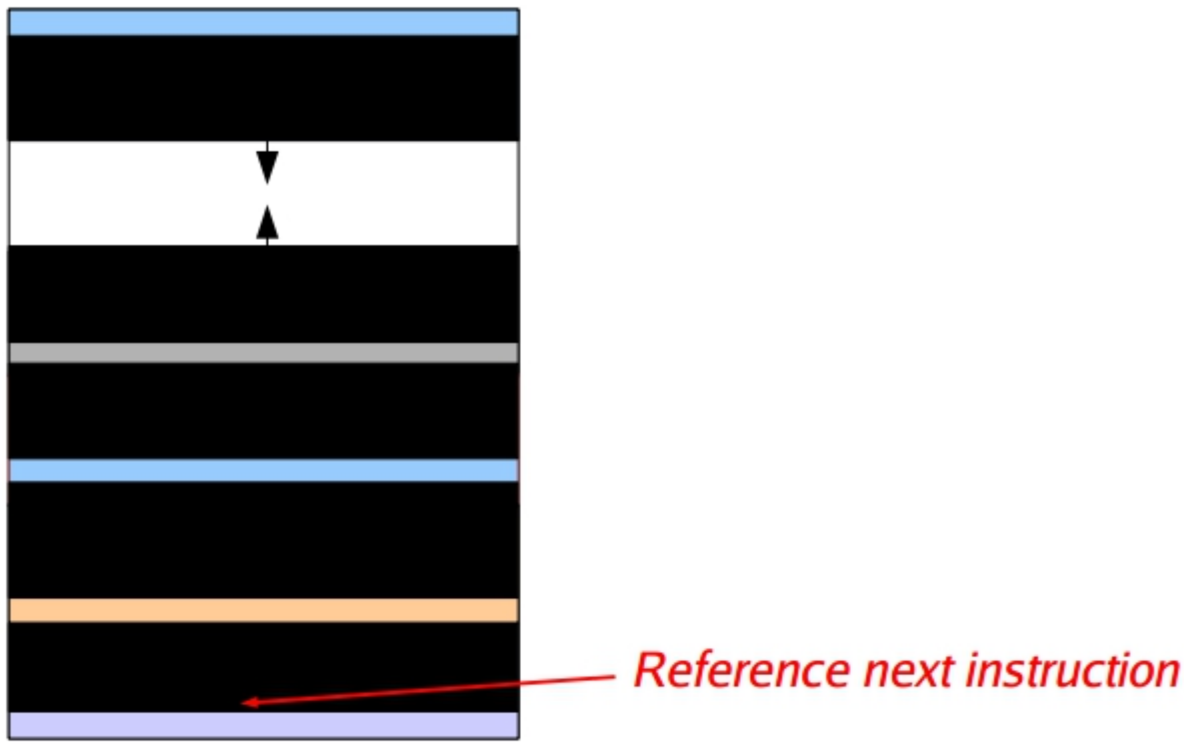
Starting up a process

- What does a process's address space look like when it first starts up?



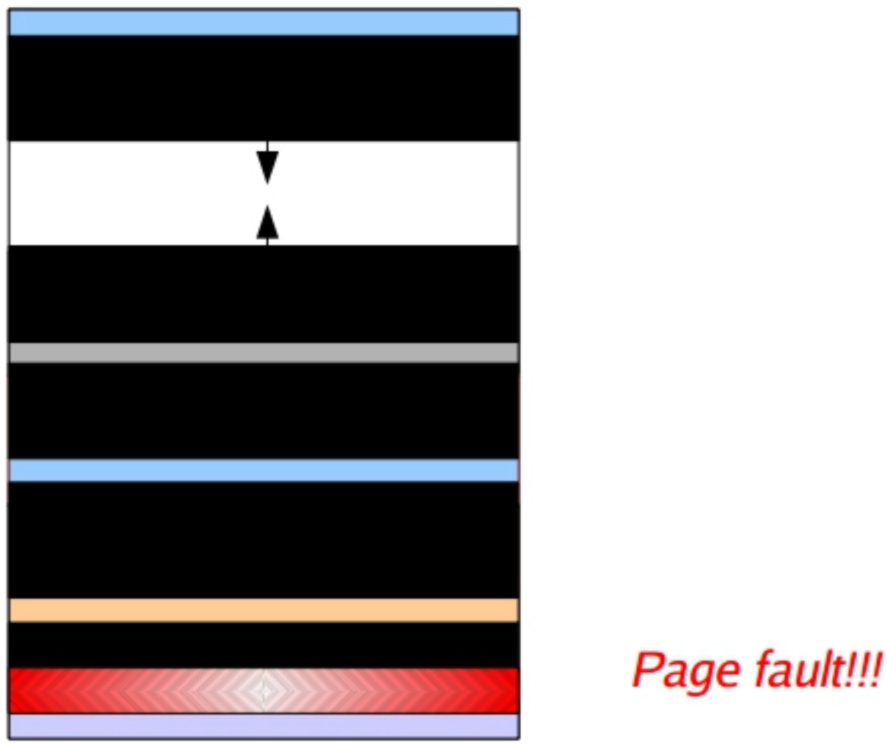
Starting up a process

- What does a process's address space look like when it first starts up?



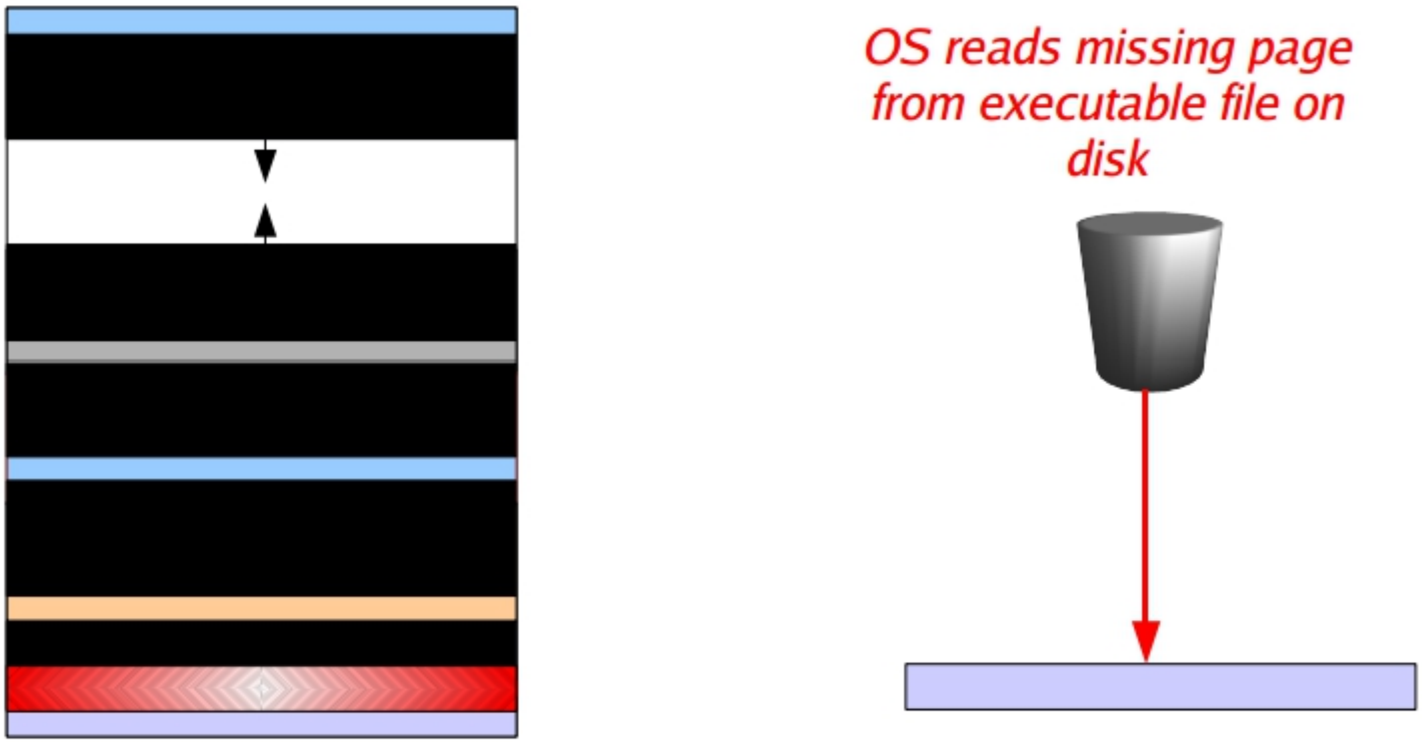
Starting up a process

- What does a process's address space look like when it first starts up?



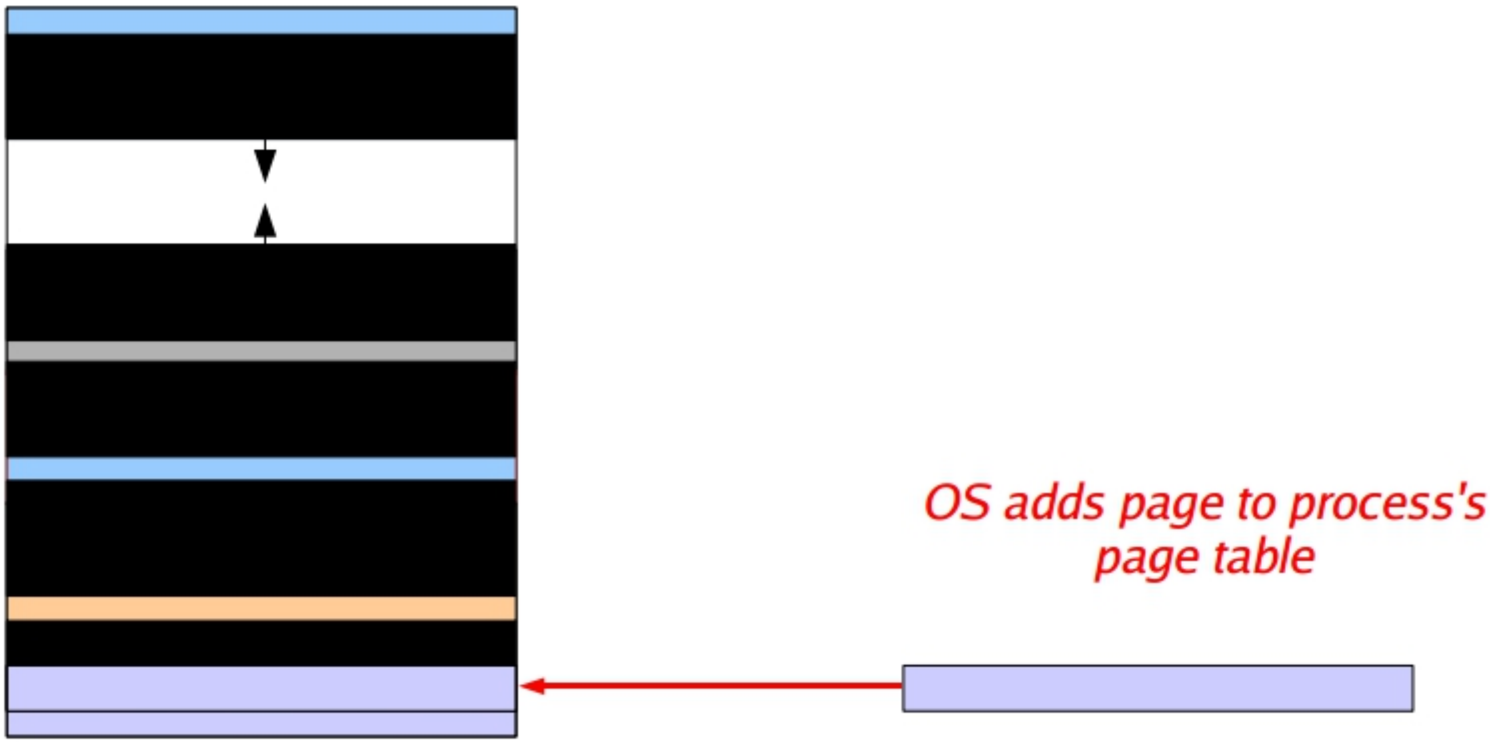
Starting up a process

- What does a process's address space look like when it first starts up?



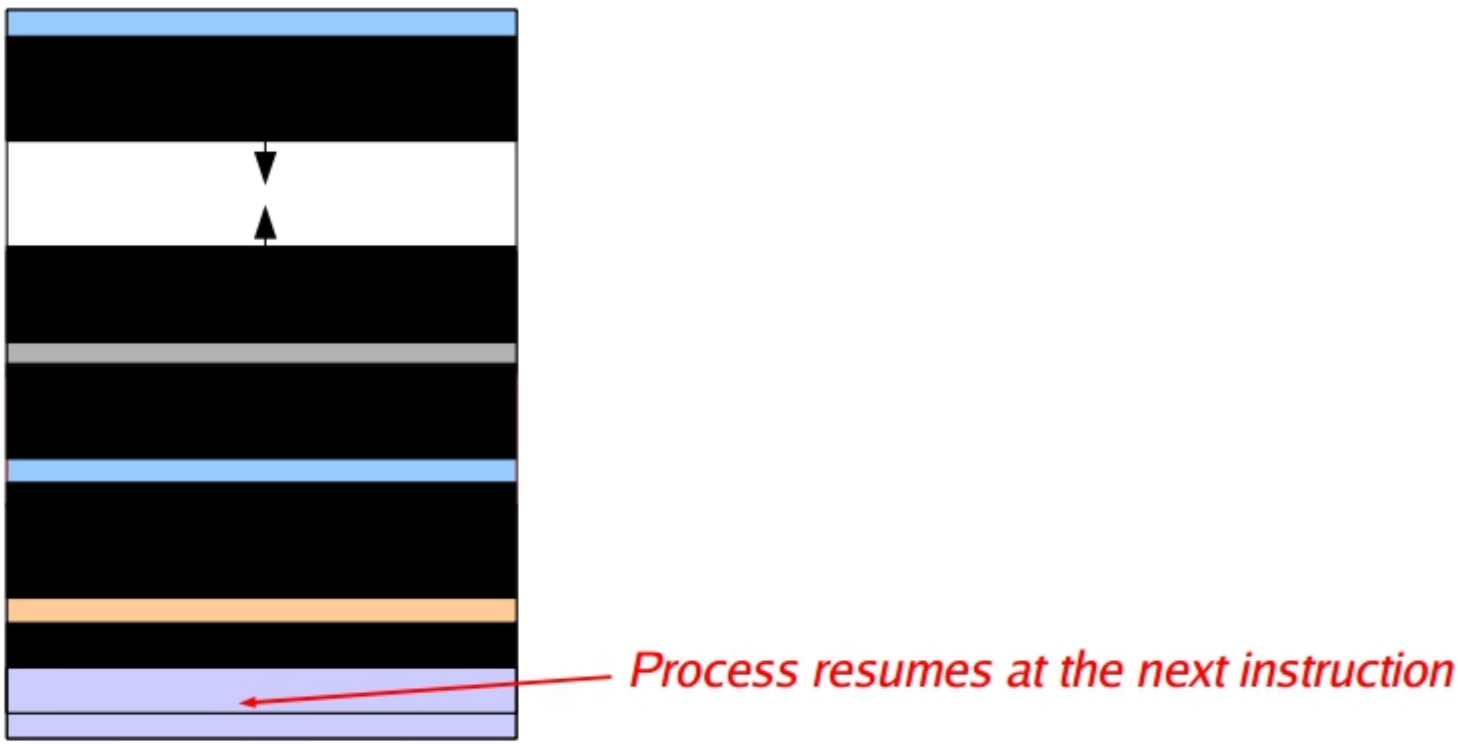
Starting up a process

- What does a process's address space look like when it first starts up?



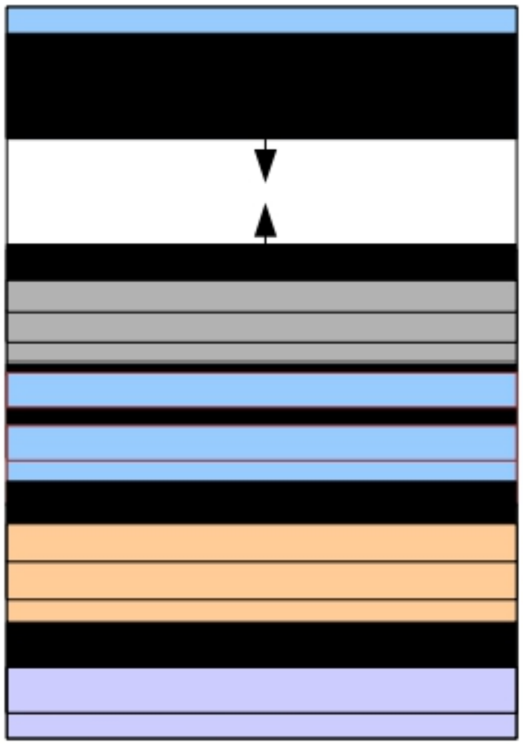
Starting up a process

- What does a process's address space look like when it first starts up?



Starting up a process

- What does a process's address space look like when it first starts up?



Over time, more pages are brought in from the executable as needed

Fetch Policy 2: Pre-paging

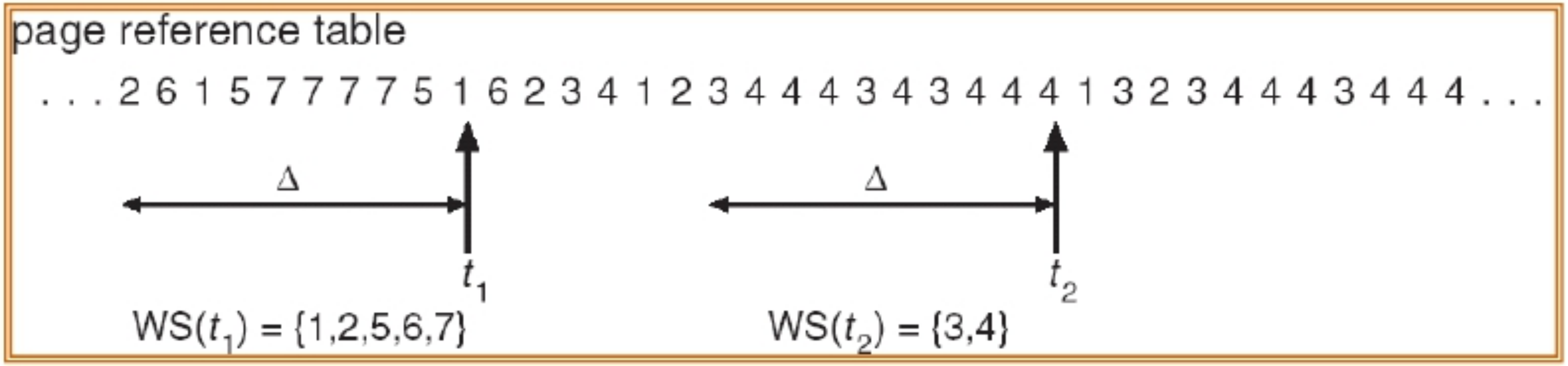
- *Pre-paging* – load pages before process runs
  - Need a ***working set*** of pages to load on context switches

- Few systems use pure demand paging, but instead, do pre-paging

Working Set Model

- ***Working set*** – A set of pages that a process is currently using
  - If entire working set is in memory, no page faults!
  - If insufficient space for working set, *thrashing* may occur.
    - Thrashing: processes busy swapping pages in and out, i.e., processes spending more time paging than executing.
- Goal: keep most of working set in memory to minimize the number of page faults

How Big is the Working Set



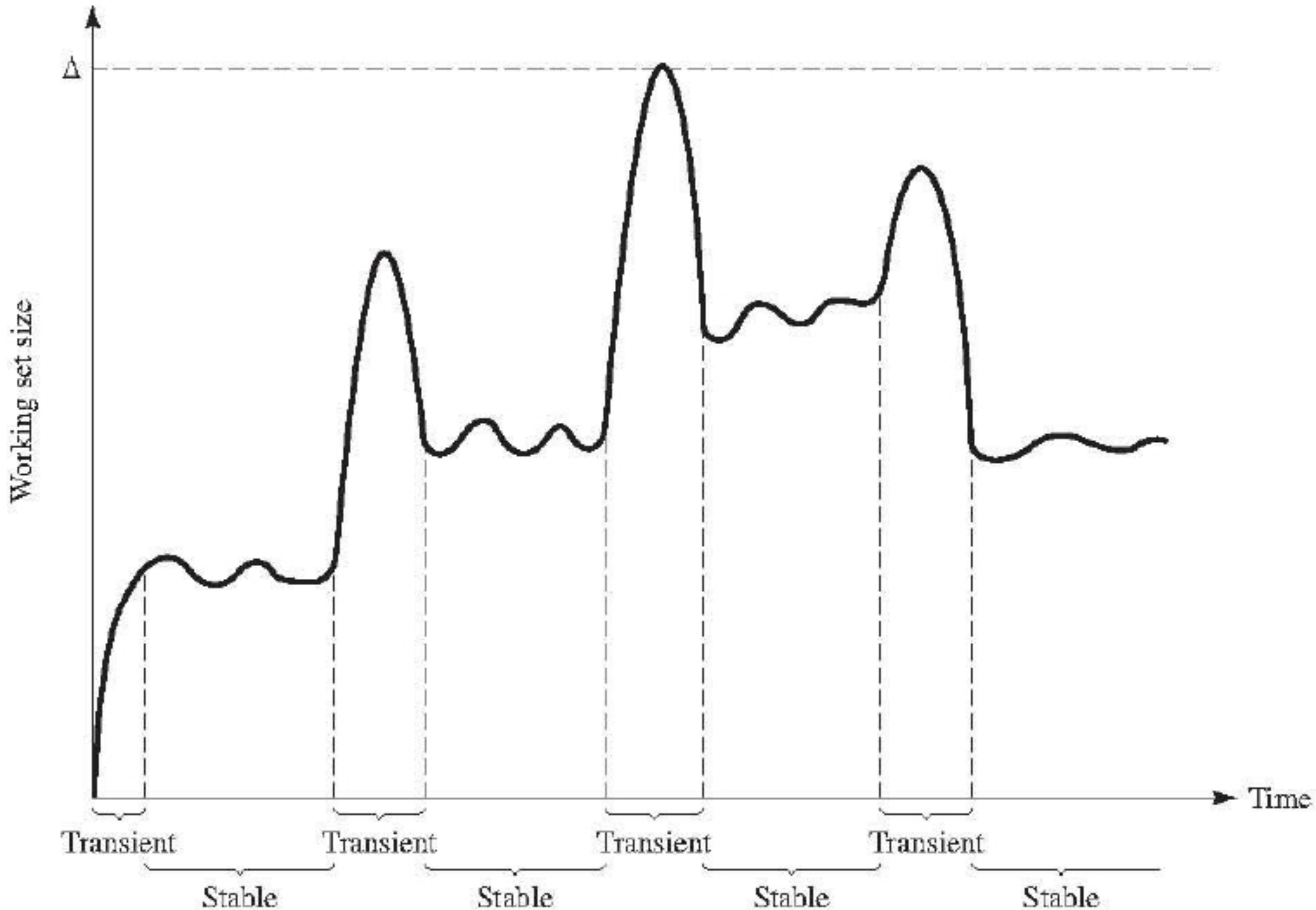
Examples of working set for K = 10

- Working set: the set of pages used by the *k most recent m*emory references
- *w(k,t)* is the size of the working set at time *t*
- Working set will change over time
- Size of working set can change over time as well

Working Set as Defined by Window Size

Sequence of Page References	Window Size, Δ 2	3	4	5
24	24	24	24	24
15	24 15	24 15	24 15	24 15
18	15 18	24 15 18	24 15 18	24 15 18
23	18 23	15 18 23	24 15 18 23	24 15 18 23
24	23 24	18 23 24	•	•
17	24 17	23 24 17	18 23 24 17	15 18 23 24 17
18	17 18	24 17 18	•	18 23 24 17
24	18 24	•	24 17 18	•
18	•	18 24	•	24 17 18
17	18 17	24 18 17	•	•
17	17	18 17	•	•
15	17 15	17 15	18 17 15	24 18 17 15
24	15 24	17 15 24	17 15 24	•
17	24 17	•	•	17 15 24
24	•	24 17	•	•
18	24 18	17 24 18	17 24 18	15 17 24 18

Typical Working Set Size



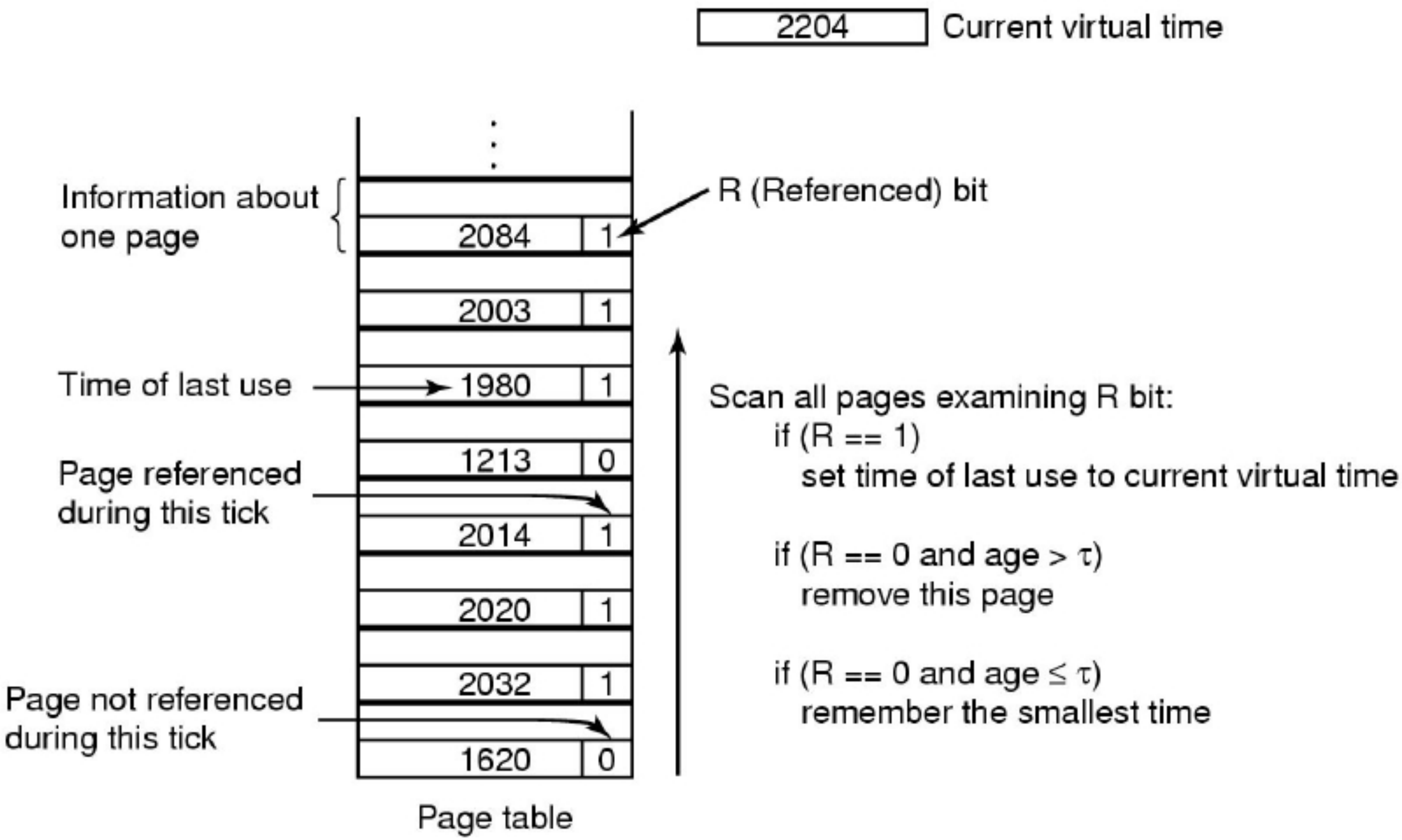
Working-Set based Page Replacement Algorithm

- If reference bit R=0, page is a candidate for removal
  - Calculate *age* = (current time – time of last use)
  - If age > threshold, page is replaced
  - If age < threshold, still in working set, but may be removed if it is oldest page in working set

- If R=1, set time of last use = current time
  - Page was recently referenced, so in working set

- If no page has R=0, choose random page for removal (one that requires no writeback)

Working Set Algorithm



Working Set Clock (WSClock) Algorithm

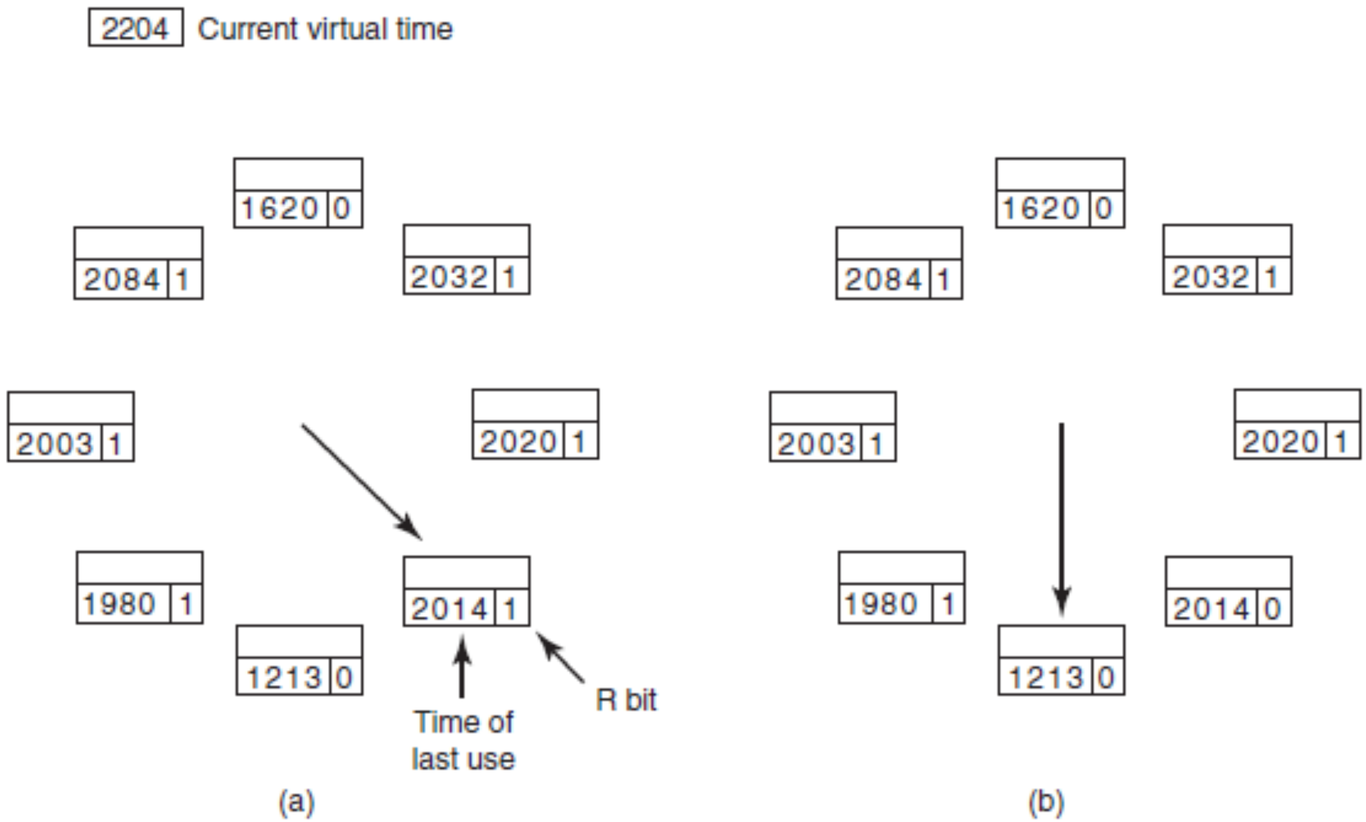
- Motivation: the basic working set algorithm needs to scan the entire page table, which is expensive!

- Use circular list of page frames
  - If R=0 and age > threshold
    - Page is clean (M=0), replace
    - Page is dirty (M=1), schedule write but advance hand to check other pages

  - If R=1, set R=0 and advance hand

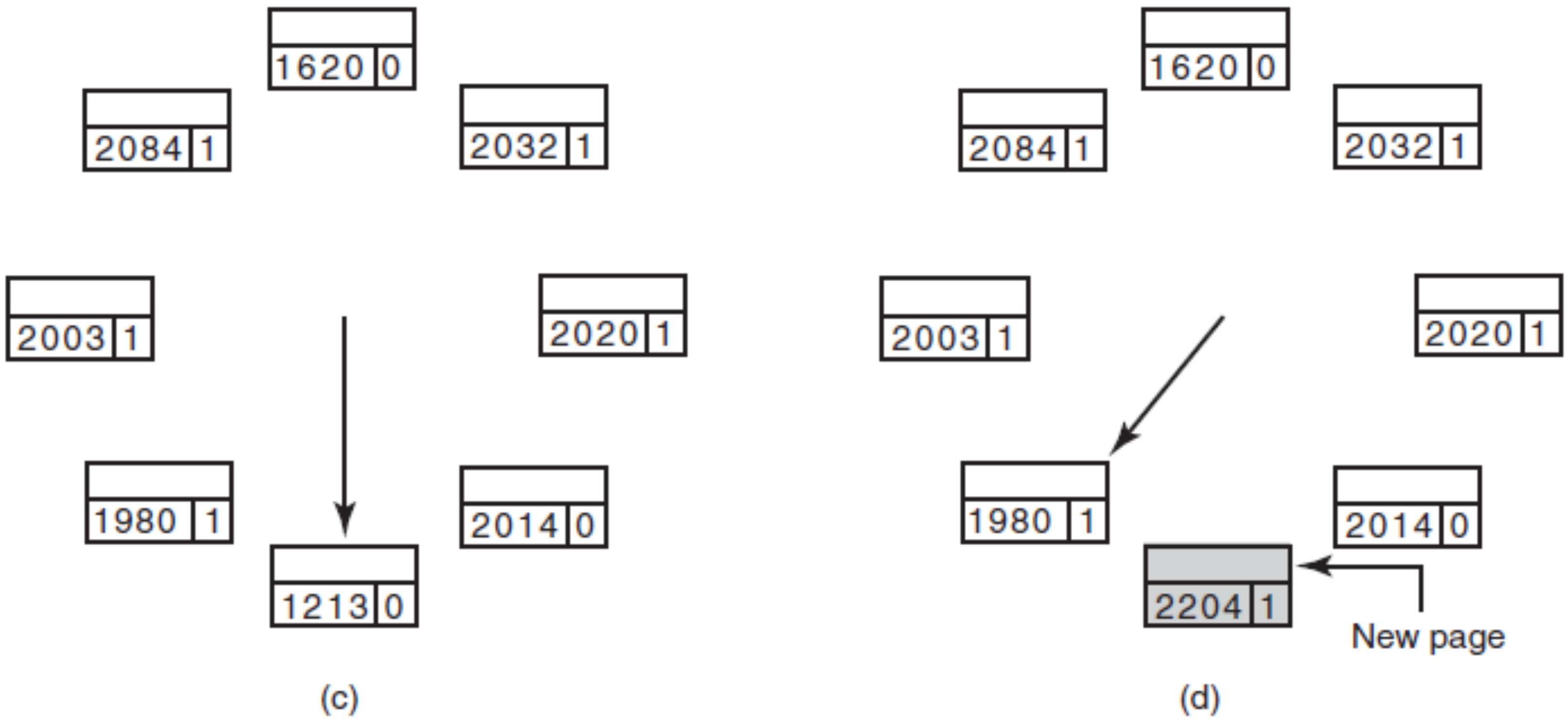
  - At end of 1st pass, if no page has been replaced:
    - If write has been scheduled, keep moving hand until write is done and page is clean. Evict 1st clean page
    - If no writes scheduled, claim any clean page even though it is in the working set

The WSClock Page Replacement Algorithm



Operation of the WSClock algorithm. (a) and (b) give an example of what happens when R = 1(set R=0 and advance hand).

The WSClock Page Replacement Algorithm



Operation of the WSClock algorithm. (c) and (d) give an example of R = 0 (Page is clean (M=0), replace)

Question

Suppose that the WSClock page replacement algorithm uses a τ of two ticks, and the system state is the following:

Page	Time stamp	V	R	M
0	6	1	0	1
1	9	1	1	0
2	9	1	1	1
3	7	1	0	0
4	4	0	0	0

where the three flag bits *V*, R, and *M* stand for Valid, Referenced, and Modified, respectively.

(a) If a clock interrupt occurs at tick 10, show the contents of the new table entries. Explain. (You can omit entries that are unchanged.)

(b) Suppose that instead of a clock interrupt, a page fault occurs at tick 10 due to a read request to page 4. Show the contents of the new table entries. Explain. (You can omit entries that are unchanged.)

Review of Page Replacement Algorithms

Algorithm	Comment
Optimal	Not implementable, but useful as a benchmark
NRU (Not Recently Used)	Very crude
FIFO (First-In, First-Out)	Might throw out important pages
Second chance	Big improvement over FIFO
Clock	Realistic
LRU (Least Recently Used)	Excellent, but difficult to implement exactly
NFU (Not Frequently Used)	Fairly crude approximation to LRU
Aging	Efficient algorithm that approximates LRU well
Working set	Somewhat expensive to implement
WSClock	Good efficient algorithm