Recitation – Week 10

Pranut Jain

Announcement

- Project 2 grading notes
- Recitation slides at http://people.cs.pitt.edu/~pranut/CS1550/

Plan for today

- Not Frequently Used Page Replacement Algorithm
- …With Ageing
- Working Sets
- WS Clock Page Replacement

Not frequently used algorithm (NFU)

- Bases decision of frequency of use rather than recency
- Associate a counter with each page
- On every clock interrupt, the OS looks at each page.
  - If the Reference Bit is set...
    - Increment that page's counter & clear the bit.
- The counter approximates how often the page is used.
- For replacement, choose the page with lowest counter.

Not frequently used algorithm (NFU)

- ***Problem:***
  - Some page may be heavily used
    - **---> Its counter is large**
  - The program's behavior changes
    - **Now, this page is not used ever again (or only rarely)**
  - This algorithm never forgets!
    - ***This page will never be chosen for replacement!***
  - We may want to combine frequency and recency

Modified NFU with aging

- **Associate a counter with each page**
- **On every clock tick, the OS looks at each page.**
  - Shift the counter right 1 bit (divide its value by 2)
  - If the Reference Bit is set…
    - **Set the most-significant bit**
    - **Clear the Referenced Bit**

T_1 100000 = 32
T_2 010000 = 16
T_3 001000 = 8
T_4 000100 = 4
T_5 100010 = 34

Working set page replacement

- *Demand paging*
  - Pages are only loaded when accessed
  - When process begins, all pages marked INVALID

Working set page replacement

- *Demand paging*
  - Pages are only loaded when accessed
  - When process begins, all pages marked INVALID
- *Locality of reference*
  - Processes tend to use only a small fraction of their pages

Working set page replacement

- ***Demand paging***
  - Pages are only loaded when accessed
  - When process begins, all pages marked INVALID
- ***Locality of Reference***
  - Processes tend to use only a small fraction of their pages
- ***Working Set***
  - The set of pages a process needs
  - If working set is in memory, no page faults
  - What if you can't get working set into memory?

Working set page replacement

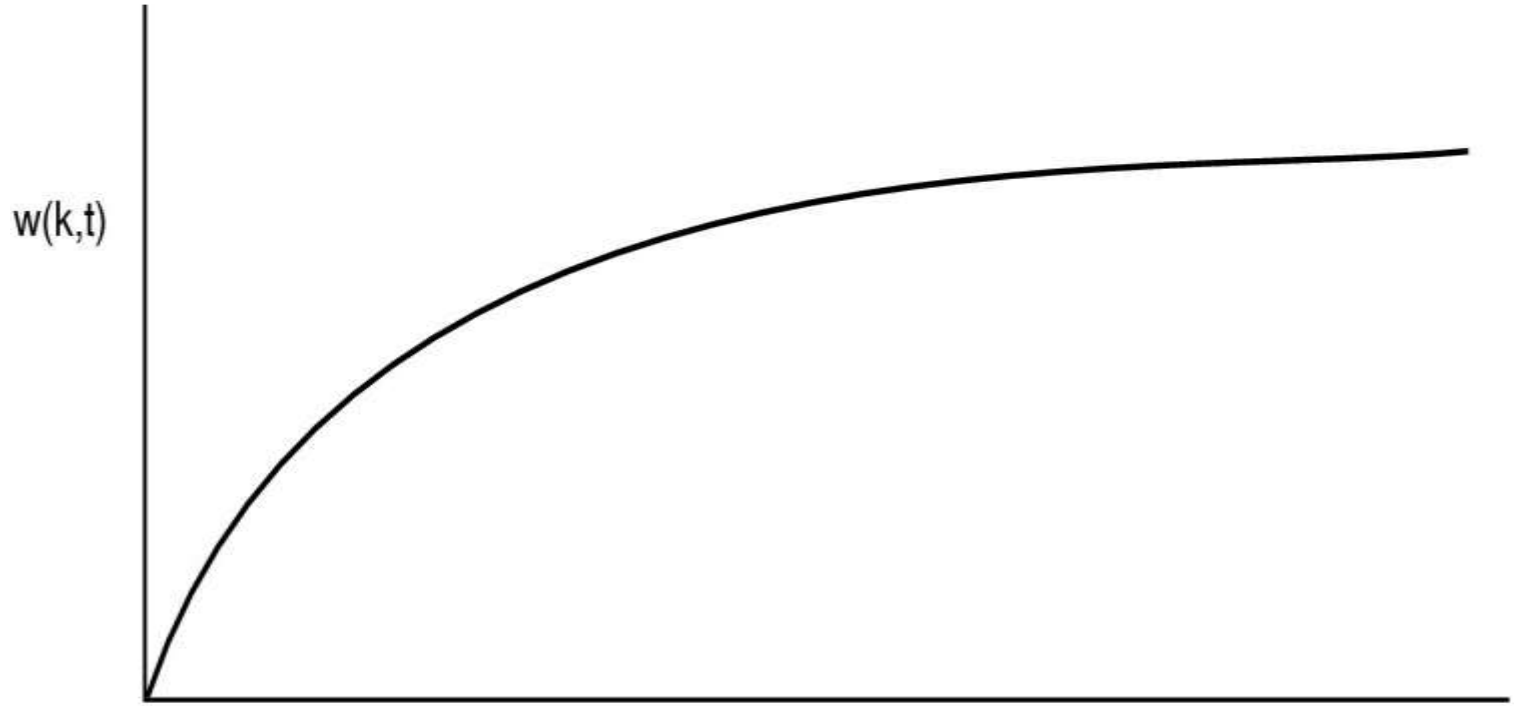
- *Thrashing*
  - If you can't get working set into memory page faults occur every few instructions
  - Little work gets done
  - Most of the CPU's time is going on overhead

Working set page replacement

- ***Based on prepaging (prefetching)***
  - Load pages before they are needed
- ***Main idea:***
  - Try to identify the process's "working set"
- ***How big is the working set?***
  - Look at the last K memory references
  - As K gets bigger, more pages needed.
  - In the limit, all pages are needed.

Working set page replacement

- ***The size of the working set:***



Working set page replacement

- **Idea:**
  - Look back over the last T msec of time
  - Which pages were referenced?
    - **This is the working set.**
- ***Current Virtual Time***
  - Only consider how much CPU time this process has seen.
- ***Implementation***
  - On each clock tick, look at each page
  - Was it referenced?
    - **Yes: Make a note of Current Virtual Time**
  - If a page has not been used in the last T msec,
    - **It is not in the working set!**
    - **Evict it; write it out if it is dirty.**

Working set algorithm

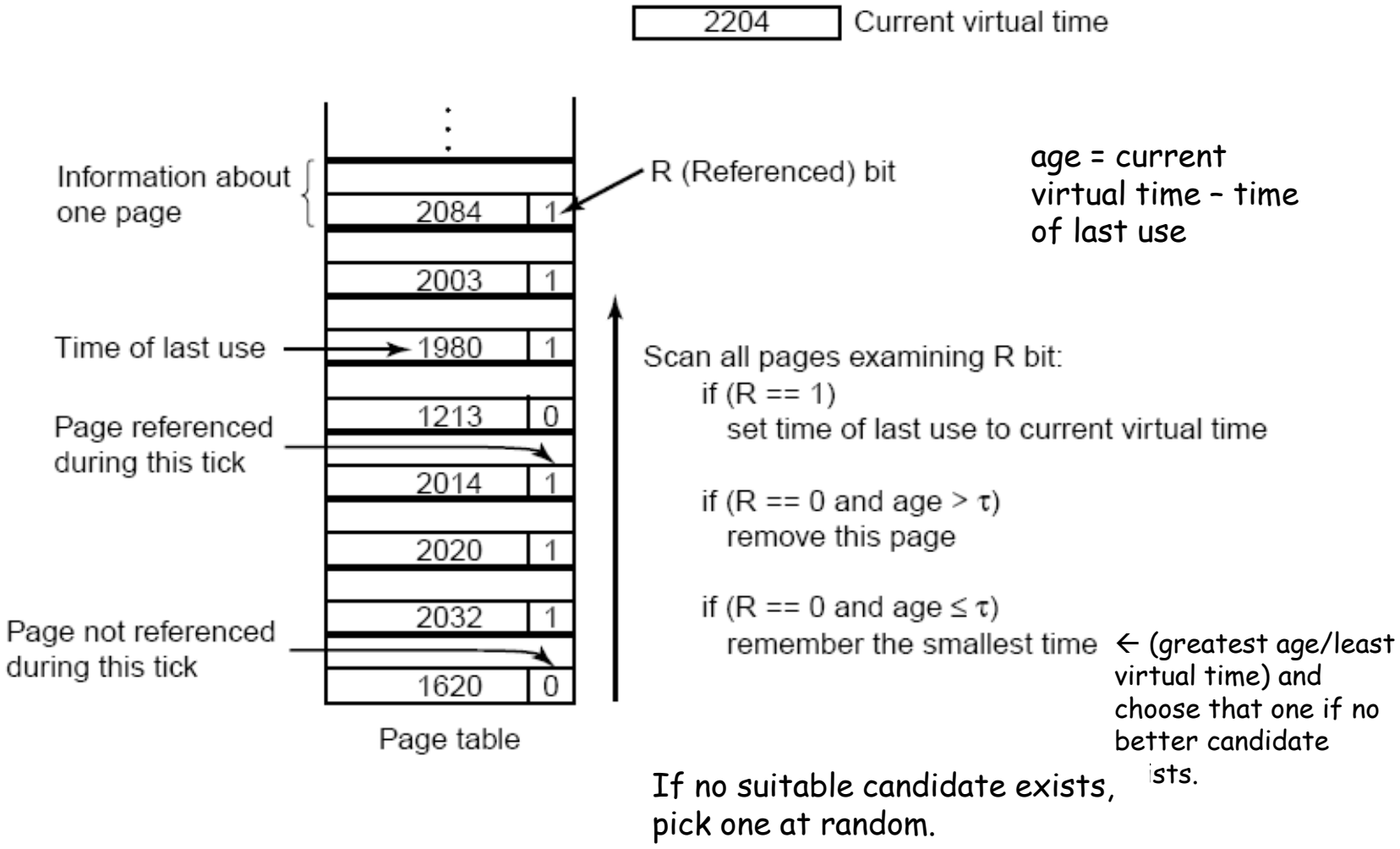


Fig. 4-21. The working set algorithm

WSClock page replacement algorithm

- An implementation of the working set algorithm
- All pages are kept in a circular list (ring)
- As pages are added, they go into the ring
- The "clock hand" advances around the ring
- Each entry contains "time of last use"
- Upon a page fault...
  - If Reference Bit = 1...
    - Page is in use now. Do not evict.
    - Clear the Referenced Bit.
    - Update the "time of last use" field.

WSClock page replacement algorithm

- **If Reference Bit = 0**
  - If the age of the page is less than T…
    - **This page is in the working set.**
    - **Advance the hand and keep looking**
  - If the age of the page is greater than T…
    - **If page is clean**
      - **Reclaim the frame and we are done!**
    - **If page is dirty**
      - **Schedule a write for the page**
      - **Advance the hand and keep looking**

WSClock page replacement

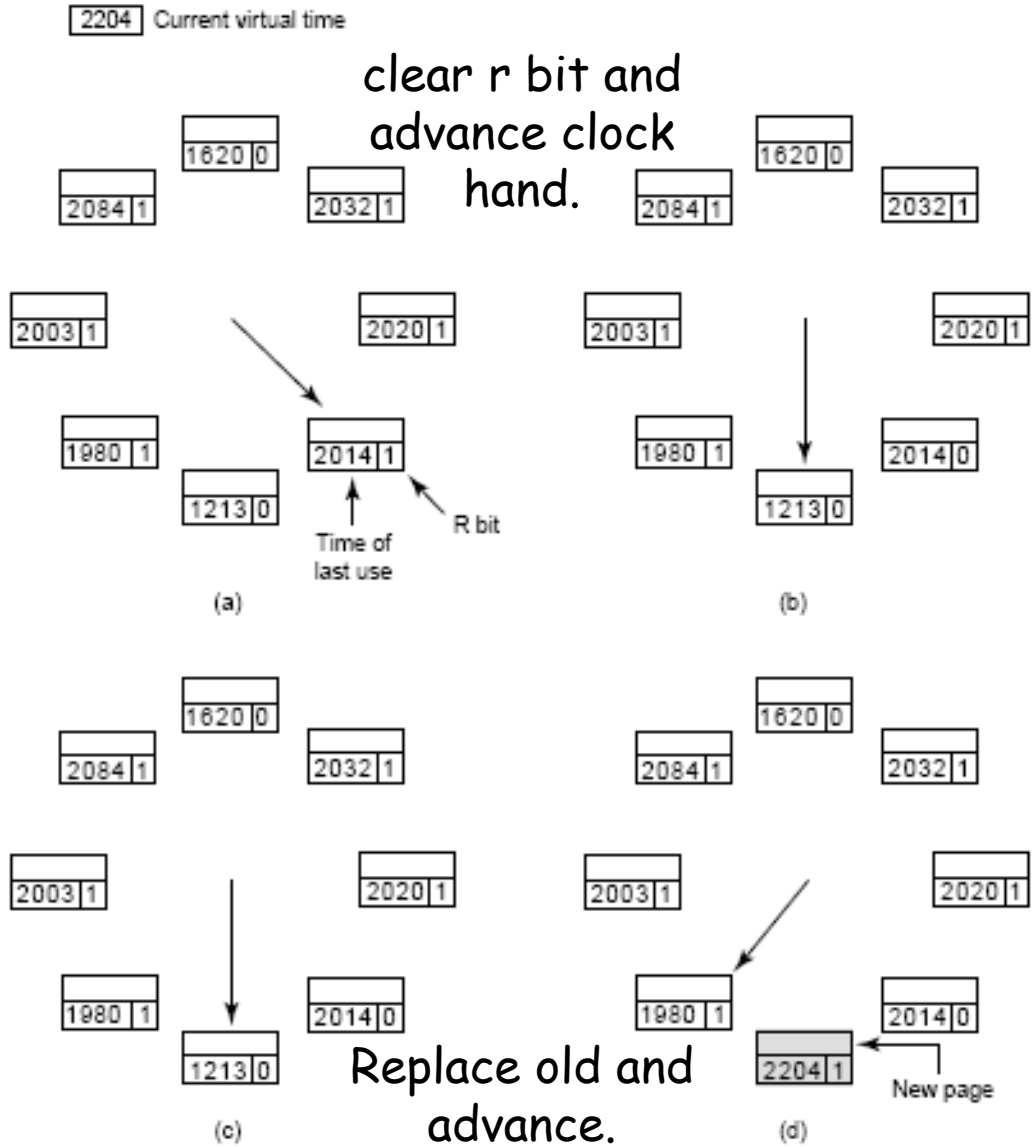


Fig. 4-22. Operation of the WSClock algorithm. (a) and (b) give an example of what happens when $R = 1$. (c) and (d) give an example of $R = 0$.