Comp 204: Computer Systems and Their Implementation

Lecture 16: Page Replacement

Today

- Virtual Memory
- Page Replacement
  - Working set model
  - Page replacement policies

Working Set Model

- The working set (Denning, 1968) of a process is defined to be the set of resources (pages) *W(T,s)* defined in the time interval *[T, T+s]*
- By the principle of locality,

$$W(T,s) \approx W(T-s,s)$$

- Hence, for each process:
  - try to ensure its working set is in memory
  - estimate working set by recording set of pages referenced in preceding time interval

Question

- Consider the following sequence of page references in a paged memory management system:

page	**p**	**q**	**r**	**q**	**q**	**q**	**p**	**r**	**r**	**q**
time	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**

- What is the working set expressed as W(3,4)?

a) q
b) r
c) qr
d) pq
e) pqr

Answer: d
pq

Question

- Consider the following sequence of page references in a paged memory management system:

page	**p**	**q**	**r**	**q**	**q**	**q**	**p**	**r**	**r**	**q**
time	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**

- What would be the predicted working set expressed as W(10,3)?

a) q
b) r
c) qr
d) pq
e) pqr

Answer: c
qr

Working Set

- The accuracy of the working set depends upon its size
  - If set too small will not cover entire locality
  - If set too large will cover several localities

- Over time the working set of a process will change as references to data and code sections move from one locality to another
  - Page fault rates will vary with these transitions

Related Policies

- Replacement policies for use with demand paging, based loosely on working set principles, include
  - Least Recently Used (LRU). Replace least-recently used page
  - First-In-First-Out (FIFO). Replace page longest in memory
  - Least Frequently Used (LFU). Replace page with fewest references in recent time period

Question 14

- Consider the following sequence of page references in a paged memory management system:

page	| p |	q |	r |	q |	q |	q |	p |	r |	r |	q |
time	0	1	2	3	4	5	6	7	8	9	10

- Page s arrives at time 10. Which of the following policies suggests we should throw out page p to make room for s?

I. LRU
II. LFU
III. FIFO

a) I only
b) III only
c) I and II
d) I and III
e) I, II and III

Answer: e
I, II and III

Anticipatory Paging

- Above policies not immune from thrashing
- A policy which more closely follows working set principles may require anticipatory paging
  - pages in working set are pre-fetched in anticipation of their need

Frame Allocation

- The fixed amount of free memory must be allocated amongst the various processes
  - Need to determine how many frames each process should get
- Each process will need a minimum number of pages
  - Dependent upon the architecture
- Allocation schemes
  - Equal allocation: each process gets an equal share of frames
  - Proportional allocation: allocate frames according to the size of the process
    - Could also implement proportional allocation based on process priorities

Performance Considerations

- Segmentation and paging overcome many limitations of linear store model, but…
- There is a performance hit
  - Each memory reference may require 2-3 store accesses
- Special hardware may help
  - registers to hold base address of current code and data segments may allow tables to be bypassed
  - special memory can aid fast table look-up (cache memory, associative store)

Page Size

- A large page size means a smaller page table
- However, large pages mean more wastage
  - On average, 50% of a page per process lost to internal fragmentation
- Small pages give better resolution
  - Can bring in only the code/data that is needed for working set
- However, small pages increase chances of page faults

Example: Windows XP

- Virtual memory implemented using demand paging
- Also implements clustering
  - When page fault occurs, bring in a number of additional pages following page required
- Each process has a working set minimum
  - Guaranteed number of pages in memory (e.g. 50)
- Also has a working set maximum (e.g. 345)
  - If page fault occurs and max has been reached, one of the process's own pages must be swapped out
- If free memory becomes too low, virtual memory manager removes pages from processes (but not below minimum)

End of Section

- Memory Management
  - Linear store model and its problems
  - Segmentation and paging
  - Virtual memory and page replacement

- The next section of the module will be Files and I/O