Virtual Memory

- Reading: OS Concepts: Chapter 10
- Contents
  - Introduction
  - Demand Paging
  - Page Replacement
  - Frame Allocation

What is virtual memory?

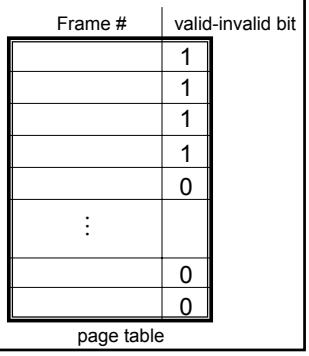
- Complete separation of logical and physical memory
  - _____
  - _____
  - _____
  - _____
- Virtual memory can be implemented via:
  - Demand paging
  - Demand segmentation

Demand paging

- Technique
  - _____
- Advantages & disadvantages
  - _____
  - _____
  - _____
- There are 3 possibilities when a page is referenced
  - If it is in-memory: _____
  - If it is not in-memory: _____
  - If it is invalid: _____

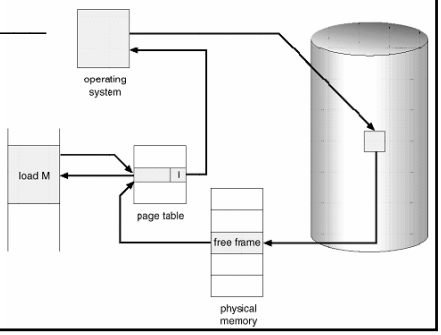
Handling memory references

- So first we must test if the page is legal
- Then check if it is present
  - Using the valid-invalid bit which
    - _____
    - _____



Handling page faults

- A memory reference causes a page table lookup
- If the page is invalid a **page fault** (interrupt) occurs
- _____
  - ☞ _____
- _____
  - ☞ _____
- _____
- _____

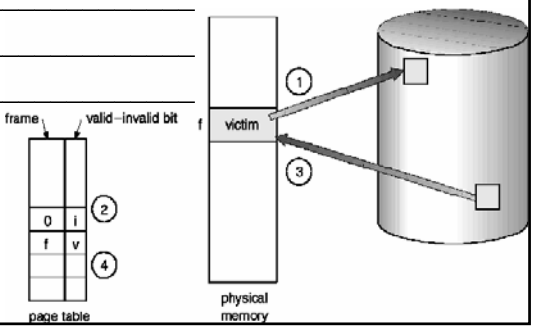


Performance

- EAT = $(1 - p)$ * EAT_{paging} + p * [Page Fault Service Time]
- p is the **Page Fault Rate**
  - p = 0.0 implies _____
  - p = 1.0 implies _____
- EAT_{paging}:_____
  _____
- Page Fault Service Time: _____
  _____
- Example
  - m = 100ns
  - ε = 10ns
  - α = 0.98
  - n = 2
  - p = 0.000001 — EAT_{paging} = _____
  - Page Fault Service Time = 25ms — EAT = _____

Page Replacement

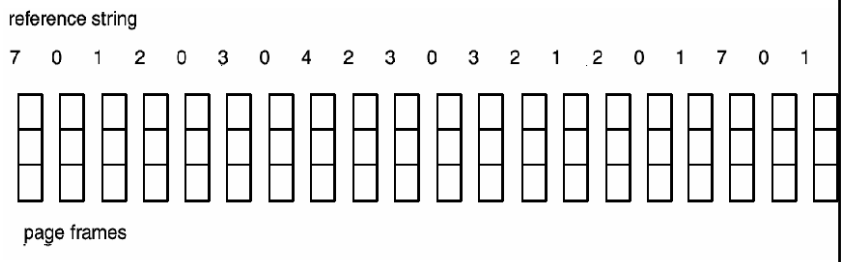
- Page replacement is necessary when there is no free memory into which to load a referenced page.
- Basic steps
  - Somehow pick a victim page
  - 1. _____
  - 2. _____
  - 3. _____
  - 4. _____



Page Replacement Issues

- The Page Fault Service Time (PFST) can be reduced if the swapped out page has not been modified
  - _____
- We also need to consider ownership of the victim
  - _____
  - _____
- We now consider possible algorithms for the replacement…
  - FIFO
  - Optimal
  - LRU

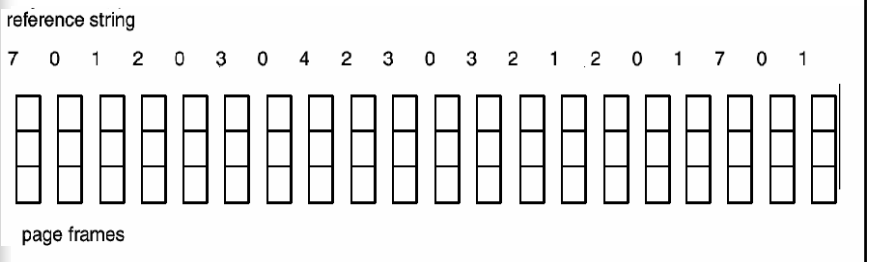
FIFO Page Replacement



- Simple concept: _____
- _____
  _____
- Belady's Anomaly: _____
  _____

9

Optimal Page Replacement



- _____
  _____
- _____
  _____

10

LRU Implementations

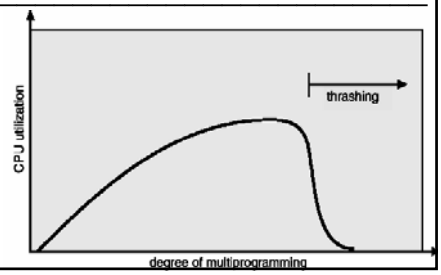
- Counter implementation
  - _____
  - _____
    _____
  - _____
    _____
- Stack implementation
  - _____
  - _____
    _____
  - _____
    _____

LRU Approximations

- Additional Reference bits algorithm
  - _____
  - _____
  - _____
  - _____
    _____
- Second chance algorithm
  - _____
  - _____
  - _____
    _____
    _____

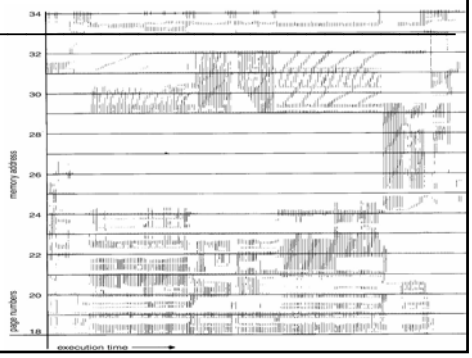
Thrashing

- Thrashing is when a process is busy (i.e. spends most of its time) swapping pages in and out
- _____
- _____



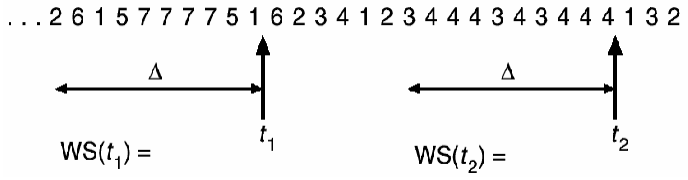
Locality model

- Paging works because the order of memory accesses are not random.
- _____
- So why does thrashing occur then?



Solving Thrashing 1: Working Set Model

- The working set of a process is the set of page references used over some time / no. of instructions Δ
  - If Δ is too small: _____
  - If Δ is too large: _____



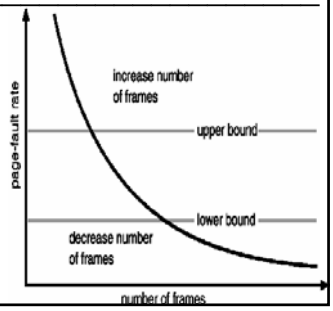
- _____
  _____
  - _____

Implementing Working Set

- It is too complex to implement the working set model as we would need to keep incredibly long lists of previous page references for each process.
- Instead we can approximate it with
  - _____
  - _____
- Example: _____
  _____
- How accurate is the approximation?
  - _____
  - _____

Solving Thrashing 2: Page Fault Frequency Scheme

- This is a simpler solution…
- Establish an 'acceptable' page fault rate
- _____
  _____
- _____
  _____



Programming Considerations

Given Array A[1024, 1024] of integer , which should you write?

Program 1
```
for (int i =0; (i<1024); i++)
  for (int j =0; (j<1024); j++)
    A[i,j] = 0;
```

Program 2
```
for (int j =0; (j<1024); j++)
  for (int i =0; (i<1024); i++)
    A[i,j] = 0;
```

Windows NT

- Uses Demand paging with clustering
  - _____
    _____
- Processes are assigned Working set minimum & maximum
  - Minimum: _____
  - Maximum: _____
- When free memory falls below a threshold
  - _____
    _____

Solaris 2

- Maintains a list of free pages
- Lotsfree _____
- Pageout process: _____
  _____
- Scanrate: _____
  _____