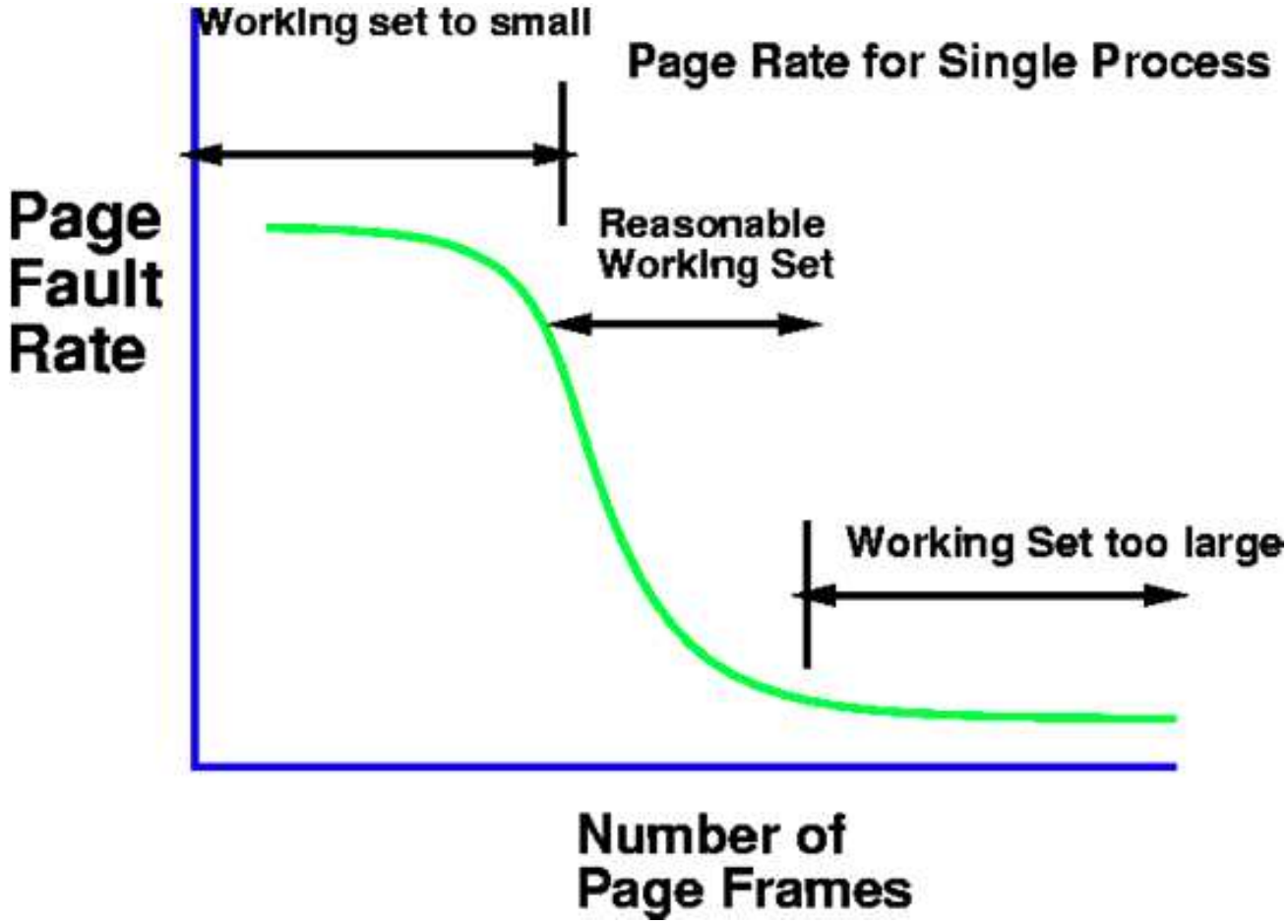
Virtual Memory (5)

Dr. Jun Zheng

CSE325 Principles of Operating Systems

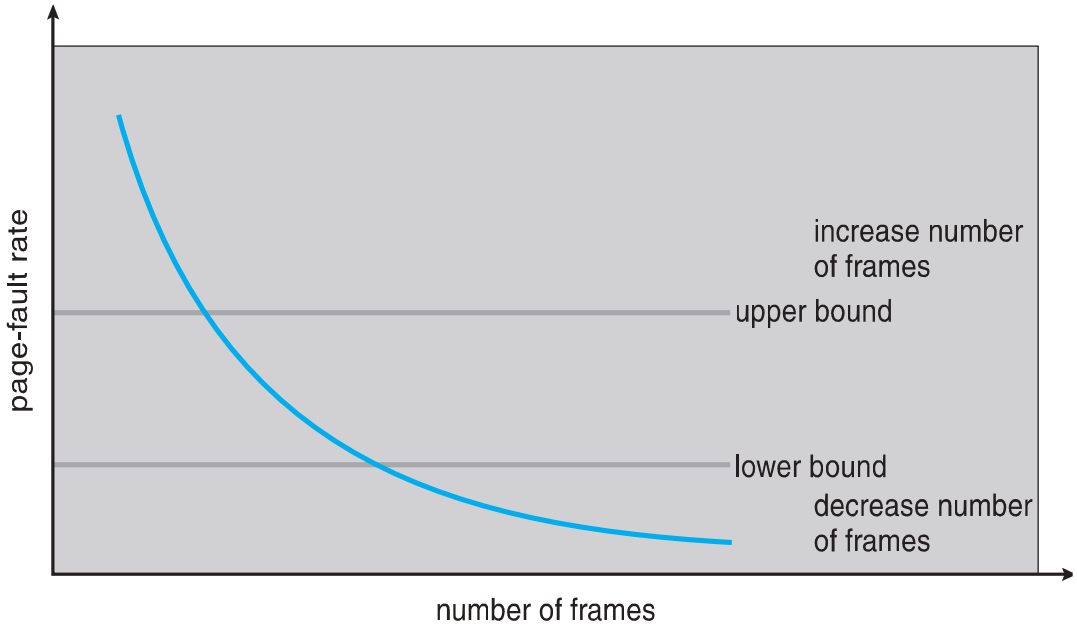
11/13/2019

Working Set Size vs. Page Fault Rate



Page-Fault Frequency

- More direct approach than working-set
- Establish "acceptable" **page-fault frequency** (**PFF**) rate and use local replacement policy
  - If actual rate too low, process loses frame
  - If actual rate too high, process gains frame



NEW MEXICO TECH
SCIENCE · ENGINEERING · RESEARCH · UNIVERSITY

Example: Windows

- Uses demand paging with **clustering**. Clustering brings in pages surrounding the faulting page
- Processes are assigned **working set minimum** and **working set maximum**
  - Working set minimum is the minimum number of pages the process is guaranteed to have in memory
  - A process may be assigned as many pages up to its working set maximum
- When the amount of free memory in the system falls below a threshold, **automatic working set trimming** is performed to restore the amount of free memory
  - Working set trimming removes pages from processes that have pages in excess of their working set minimum

Allocating Kernel Memory

- Treated differently from user memory
- Often allocated from a free-memory pool
  - Kernel requests memory for structures of varying sizes
  - Some kernel memory needs to be contiguous
    - i.e. for device I/O

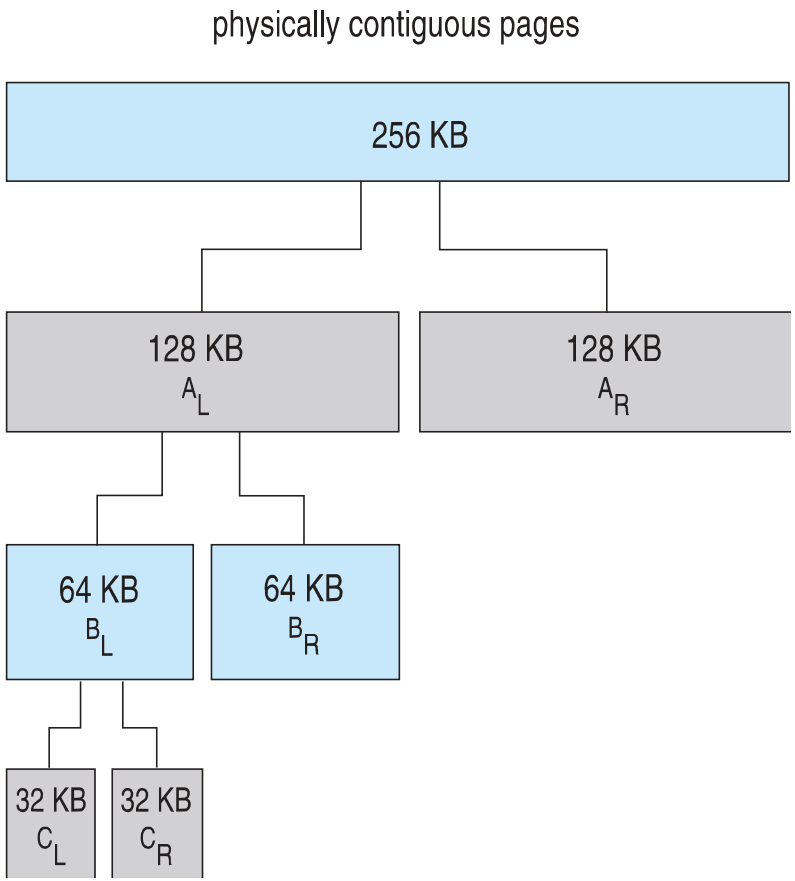
Buddy Allocation (1)

- Allocates memory from fixed-size segment consisting of physically-contiguous pages
- Memory allocated using **power-of-2 allocator**
  - Satisfies requests in units sized as power of 2
  - Request rounded up to next highest power of 2
  - When smaller allocation needed than is available, current chunk split into two buddies of next-lower power of 2
    - Continue until appropriate sized chunk available

Kenneth C. Knowlton. A Fast storage allocator. Communications of the ACM 8(10):623-625, Oct 1965

Buddy Allocation (2)

- For example, assume 256KB chunk available, kernel requests 21KB
- Advantage – quickly **coalesce** unused chunks into larger chunk
- Disadvantage - fragmentation



In-class Work 7

- Consider a system with 1MB of available memory and requests for 42KB, 396KB, 10KB, and 28KB. Show the amount of memory allocated for each request and the state of memory after each request.
- How much internal fragmentation exists in this scenario?
- How much external fragmentation exists in this scenario?

Answer

Original: 1MB

42KB -> 64KB: 64KB(x) 64KB 128KB 256KB 512KB

396KB->512KB: 64KB(x) 64KB 128KB 256KB 512KB(x)

10KB->16KB: 64KB(x) 16KB(x) 16KB 32KB 128KB 256KB 512KB(x)

28KB->32KB: 64KB(x) 16KB(x) 16KB 32KB(x) 128KB 256KB 512KB(x)

Internal fragmentation: (64-42)+(512-396)+(16-10)+(32-28) = 148KB

External fragmentation: 16+128+256 = 400KB

NEW MEXICO TECH
SCIENCE · ENGINEERING · RESEARCH · UNIVERSITY

Slab Allocation (1)

- Motivation
  - Frequent (de)allocation of cetain kernel objects
    - **e.g. file struct and inode**
  - Other allocators: overly general; assume variable size
- **Slab: cache** of slots
  - slot size = object size
  - free memory management = bitmap
  - allocate: set bit and return slot
  - Free: clear bit
- Used in FreeBSD and linux, implemented on top of buddy page allocator, for objects smaller than a page

Slab Allocation (2)