

CS323 Operating Systems Memory Management II

Yuanyuan Zhou
Lecture 14
2/21/2003

Content of this lecture

- Administrative announcements
- Storage Management
- Virtual Memory and Paging
- Summary

2

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Administrative

- MP2

3

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Review

- Memory Manager
 - Monitor used and free memory
 - Allocate memory to processes
 - Reclaim (De-allocate) memory
 - Swapping between main memory and disk
- Mono-programming memory management
 - Overlay
- Multi-programming memory management
 - Fixed partition
 - Relocation and protection

4

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Swapping

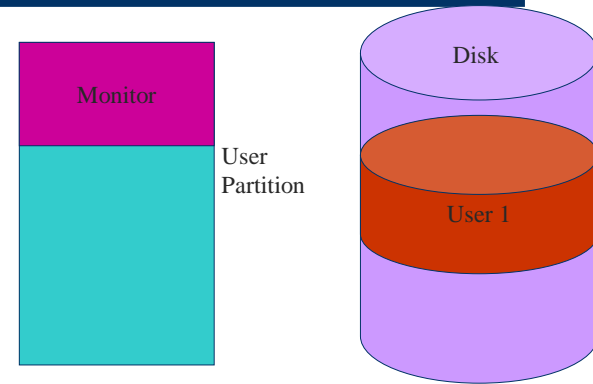
- Move a part of or the whole process to disk
- Allows several processes to share a fixed partition
- Processes that grow can be swapped out and swapped back in a bigger partition

5

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Swapping

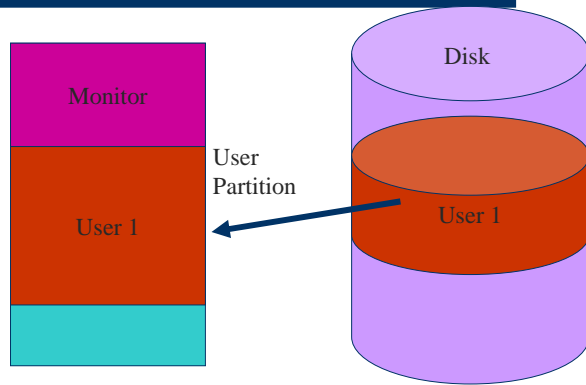


6

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Swapping

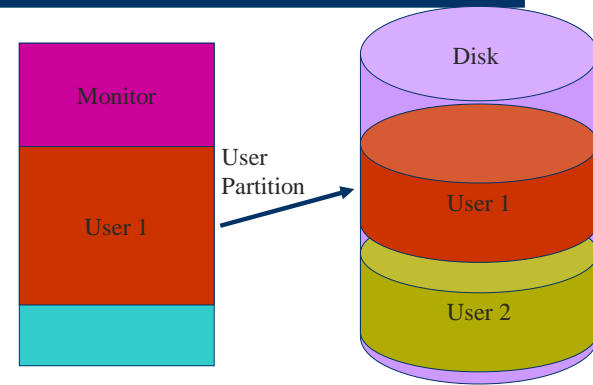


7

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

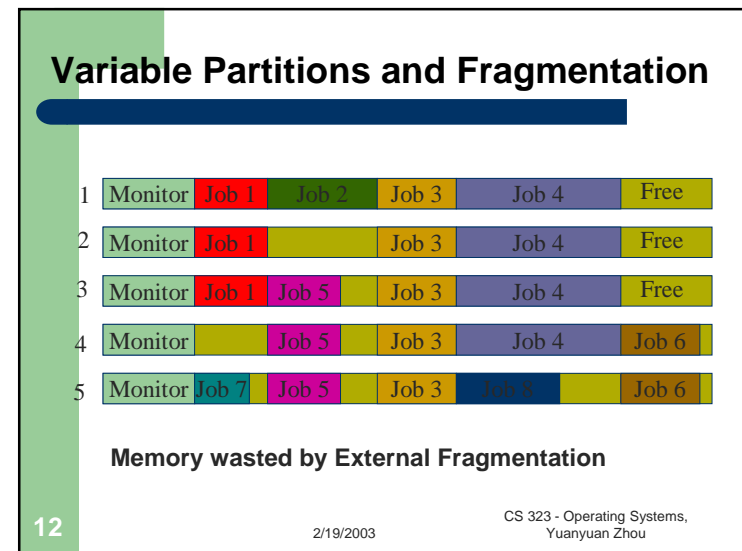
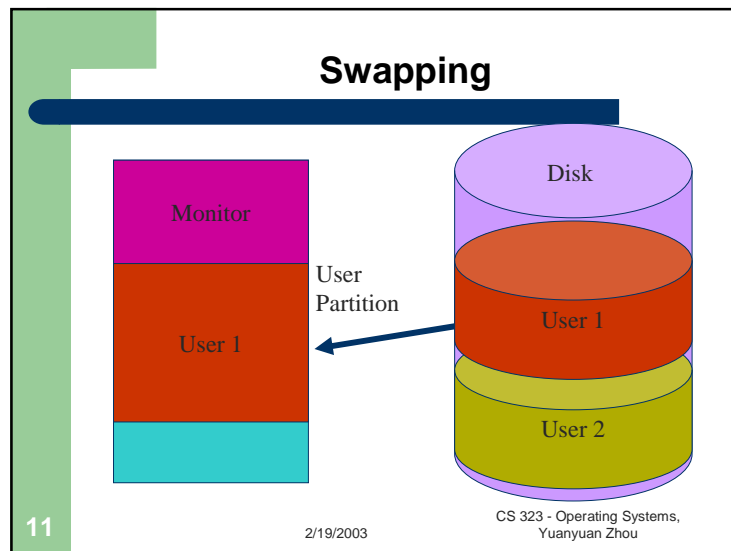
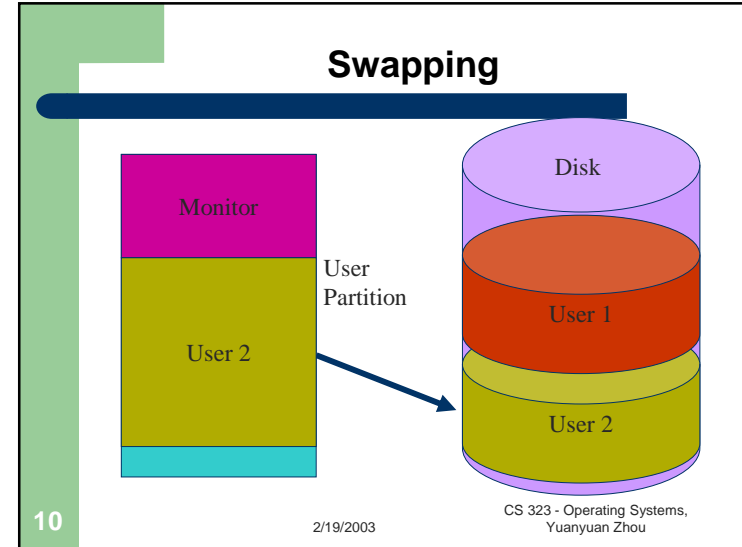
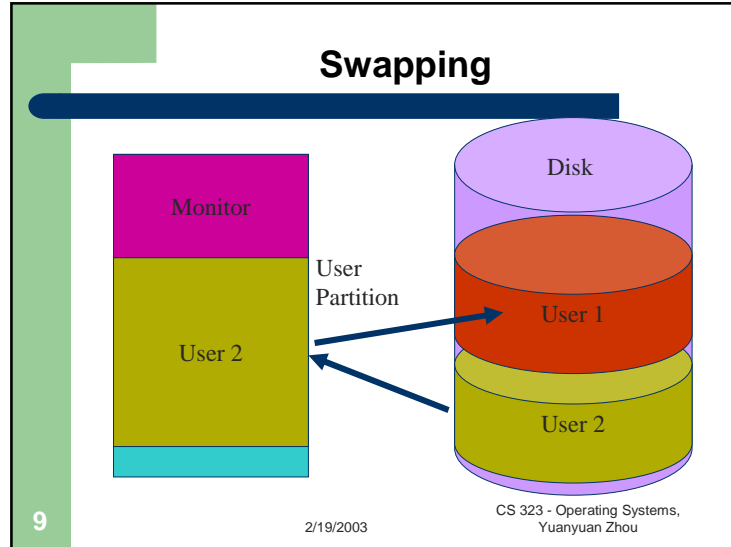
Swapping



8

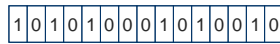
2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou



Memory Management with Bitmaps

- Use bitmaps for free lists.
- Memory is divided into allocation units.
 - One allocation unit corresponds to 1bit in the bitmap
 - 0: free, 1: allocated
- Size of allocation unit
 - The smaller the allocation unit, the larger the bitmap.
- Problem: allocation
 - When a new process arrives, the manager must find consecutive 0 bits in the map.
 - Searching a bitmap for a run of a given length is a slow operation.



13

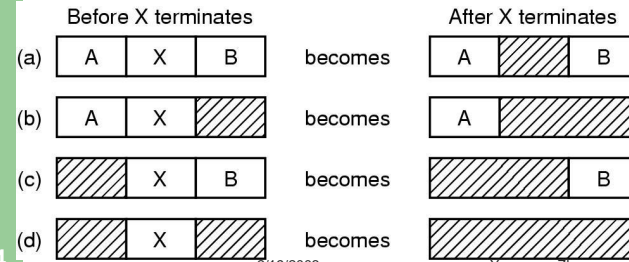
2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Memory Management with Linked Lists

- Use a linked list of allocated and free memory segments (called hole)
 - sorted by the address or by the size

Four neighbor combinations for the terminating process X



14

2/19/2003

Yuanyuan Zhou

Storage Placement Strategies

- **Analogy: Shoe Fitting**
 - Size: 6-11
- **Best Fit**
 - Use the hole whose size is equal to the need, or if none is equal, the hole that is larger but closest in size.
 - Problem: Creates small holes that can't be used.
- **First Fit**
 - Use the first available hole whose size is sufficient to meet the need.
 - Problem: Creates average size holes.
- **Next Fit.**
 - Minor variation of first fit: search for the last hole stopped.
 - Problem: slightly worse performance than first fit.
- **Worst Fit.**
 - Use the largest available hole.
 - Problem: Gets rid of large holes making it difficult to run large programs.
- **Quick Fit.**
 - maintains separate lists for some of the more common sizes requested.
 - When a request comes for placement it finds the closest fit.
 - This is a very fast scheme, but a merge is expensive. If merge is not done, memory will quickly fragment in a large number of holes into which no processes fit.

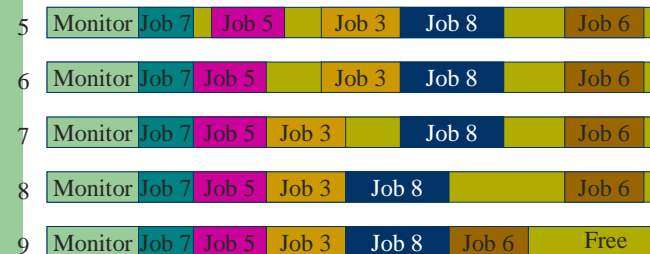
15

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Compaction

- Assumes programs are all relocatable
- Processes must be suspended during compaction
- Need be done only when fragmentation gets very bad



16

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Multiple Base Registers

- Break programs into smaller units because they will fit better
- Use multiple base registers, one for each unit
- Examples
 - Code/Data
 - Constants/variables

17

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Storage Management Problems

- Fixed partitions suffer from internal fragmentation
- Variable partitions suffer from external fragmentation
- Compaction suffers from overhead
- Partitions must be less in size than real memory
- Overlays are painful to program efficiently
- Swapping requires writing to disk sectors

18

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

How Bad Is Fragmentation?

- Statistical arguments - Random sizes
- First-fit
- Given N allocated blocks
- $0.5 * N$ blocks will be lost because of fragmentation
- Known as 50% RULE

19

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Virtual Memory

- Provide user with virtual memory that is as big as user needs
- Store virtual memory on disk
- Cache parts of virtual memory being used in real memory
- Load and store cached virtual memory without user program intervention

20

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Why Use Virtual Memory

- Group Discussion (2 minutes)

21

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Benefits of Virtual Memory

- Use secondary storage(\$)
 - Extend DRAM(\$\$\$) with reasonable performance
- Protection
 - Programs do not step over each other
- Convenience
 - Flat address space
 - Programs have the same view of the world
 - Load and store cached virtual memory without user program intervention
- Reduce fragmentation:
 - make cacheable units all the same size (page)
- Remove memory deadlock possibilities:
 - permit pre-emption of real memory

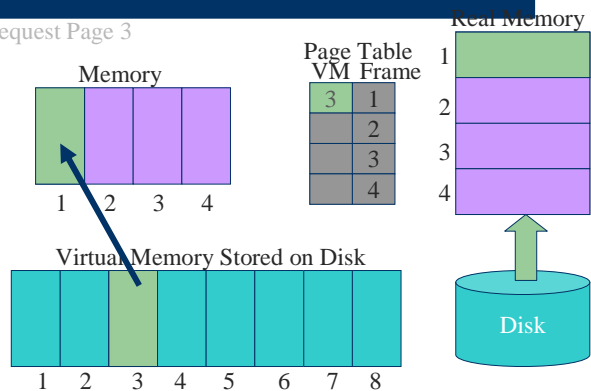
22

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Paging Request

Request Page 3



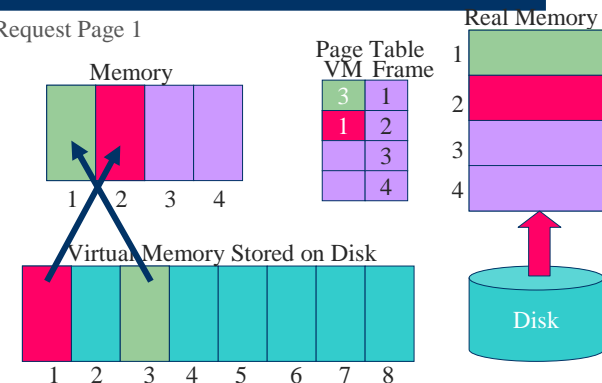
23

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Paging

Request Page 1

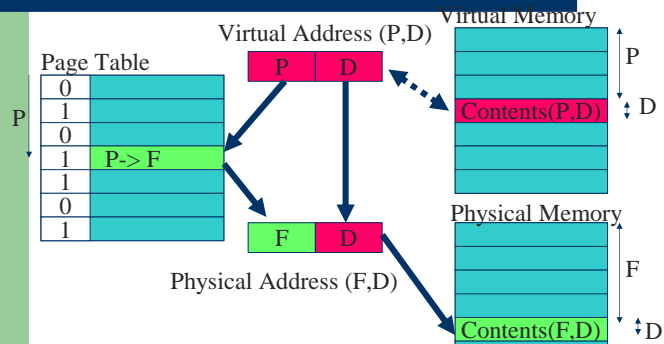


24

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Page Mapping Hardware

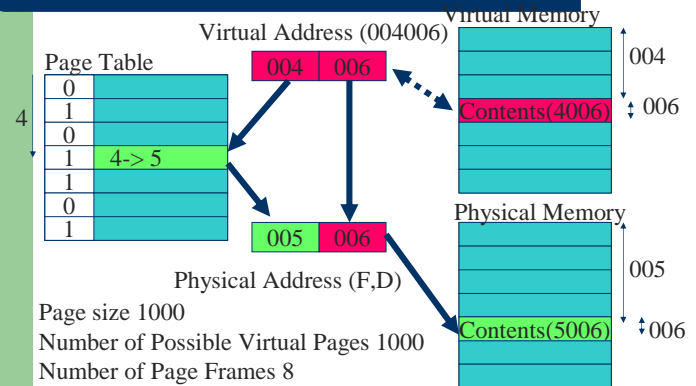


29

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Page Mapping Hardware



30

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Reminder

- MP2

31

2/19/2003

CS 323 - Operating Systems,
Yuanyuan Zhou