Discussion Week 6

TA: Kyle Dewey

Overview

- Project #2 questions

- Project #2 test cases

- Exam sample questions

- Project #3 high level overview

Project #2 Error Conditions

- Do not worry about error codes

- Still should check for error conditions for your own testing

Project #2 Questions?

Project #2 Test Cases

- In `code/test`

- These will be used to test your code, along with similar variants of them

- Altogether, they have wide coverage

Some Cases

- `fork.c`

- `ncat.c`

- `test1.c`

Exam Sample Questions

Question:"Do OSes actually swap anything anymore?"

Relevance

- Modern systems have tons of memory

- Usually a lot more free memory than in use

  - Setups vary widely

Memory Usage

- Linux has the `free` command

- Shows free and used memory, along with swap

Occasional Webserver

	total	used	free	shared	buffers	cached
Mem:	3962	3782	179	0	181	3009
-/+ buffers/cache:		591	3370			
Swap:	1983	0	1983			

Occasional Fileserver

	total	used	free	shared	buffers	cached
Mem:	496	481	14	0	270	65
-/+ buffers/cache:		144	351			
Swap:	792	1	790			

CSIL Under Load

	total	used	free	shared	buffers	cached
Mem:	8072	7493	579	0	277	3292
-/+ buffers/cache:		3923	4149			
Swap:	12229	205	12023			

Server After Load Spike

	total	used	free	shared	buffers	cached
Mem:	8002	7697	304	0	2098	4652
-/+ buffers/cache:		946	7055			
Swap:	1811	353	1458			

Virtualization

- Multiple OSes feed off the same memory

- Memory tends to be the greatest limitation

- Special software used to force OS to page in one VM when another is starved for memory

Virtualized Memory

4 GB total

Needs 3 GB

OS1

4 GB total

Needs 2 GB

OS2

4 GB total

Machine

On Paging

- Paging is a last resort mechanism

- A private, lightly loaded system will likely never need it

- In enterprise environments, it ranges from very nice to absolutely vital

Project #3 Overview

Part 1

- Need to implement paging

- Tests will be so large as to need paging

- Paging algorithm cannot be naive

  - Least recently used (LRU)

  - Second chance FIFO queue

FIFO Queue

Oldest Page   Newest Page
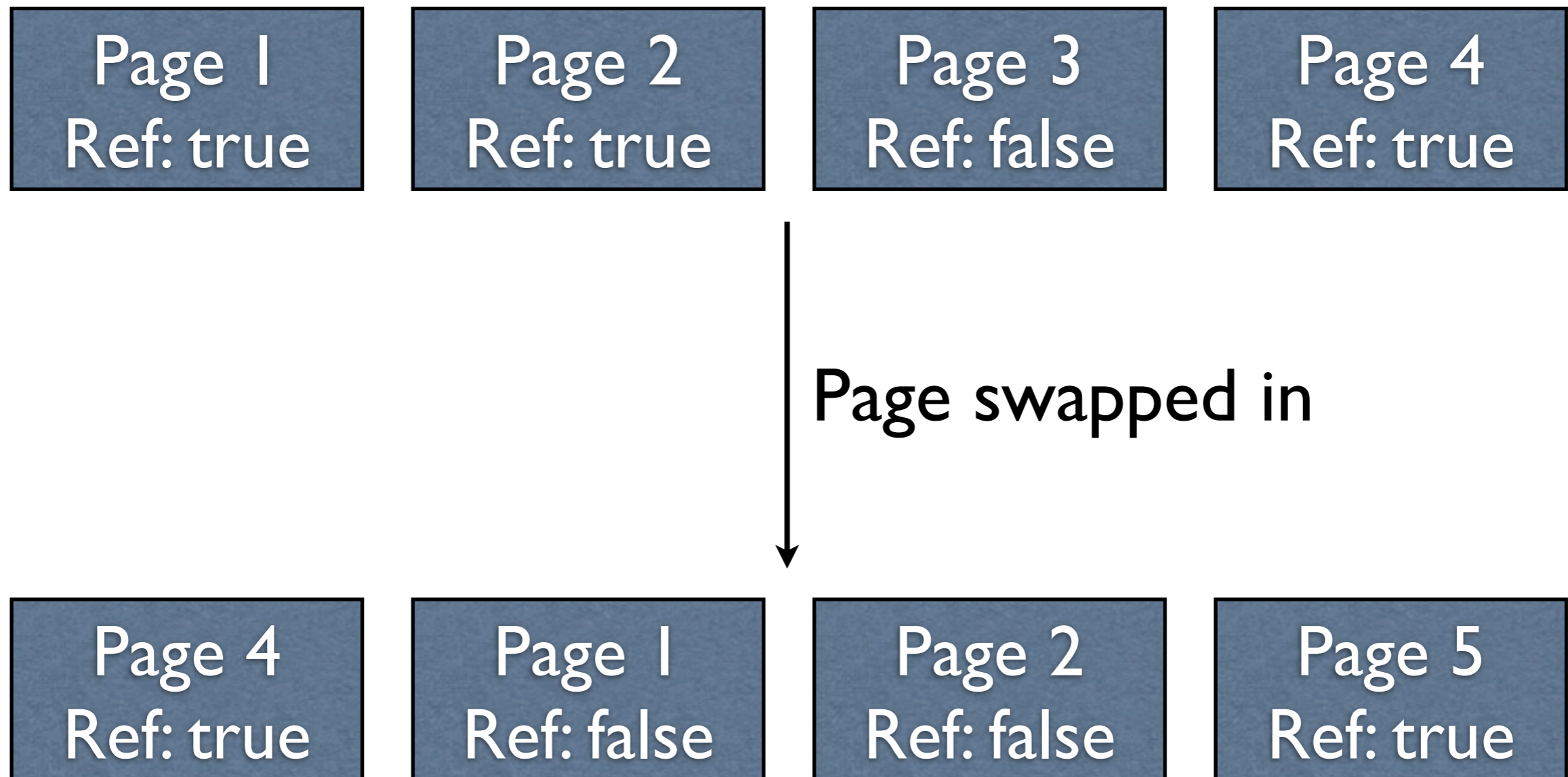
| Page 1 | Page 2 | Page 3 | Page 4 |

Page swapped in

| Page 2 | Page 3 | Page 4 | Page 5 |

Oldest Page   Newest Page

FIFO + Second Chance

Page 1 Ref: true	Page 2 Ref: true	Page 3 Ref: false	Page 4 Ref: true

Page swapped in

Page 4 Ref: true	Page 1 Ref: false	Page 2 Ref: false	Page 5 Ref: true

Part 2

- Make the NACHOS filesystem more useful

- Will be using the NACHOS disk for this

- Flat file hierarchy

Part 2

- **Implement** `Create(),Open(),`
  `Remove(),List(),Seek(),Read(),`
  `ReadAt(),Write(),WriteAt(),`**and**
  `Length()`