**CSSE2310: Fletch’s Memory and File Cheat Sheet**

File System Questions:

Unix File System - Largest File

Num entries per block = block size/ entry (pointer) size

Total blocks = directs + 1st indirects + 2nd indirects

= direct + (a \* num entries) + (b \* num entries^2)

where a = number of 1st indirect pointers on inode

and b = number of 2nd indirect pointers on inode

File size = total blocks \* size of 1 block

Unix File System - How many index blocks needed for a file

Num blocks needed = File size / block size

First 12 blocks are direct so:

index blocks needed = (file size / block size) / entries per block

Remember: an answer of 1.4, would mean 2 index block was used (its on the second block

Unix File System - Number of reads to read block b

Check if b < num direct :

If it is, only 1 read is required

Check if b < num direct + (number of 1st indirect pointers \* pointers per block):

If it is, the address only needs 1 level of indirect.

Check which index block it is on (to see if it is already in cache)

If its not in cache, 2 reads are required

Check if b < num direct + (number of 1st indirect pointers \* pointers per block)

+ (number of 2nd indirect pointers \* pointers per block^2)

If it is, check if any index blocks followed are already in cache

If none are in cache, 3 reads are required

Linked File System - Number of reads to read blocks a, b, c, d (size order)

Number of reads = highest block to read (d) + number of reads requested (4)

Virtual Memory Questions:

Memory used for page tables for process using X bytes of virtual memory (2L)

Num pages needed = total size/ page size

Num entries per L2 page = page size/ entry size

Num L2 pages needed = pages needed / num entries

Total tables needed = 1L table + 2L tables

Memory needed = total tables \* page size

Checking a page table scheme (3L)

Num entries per page = page size / entry size

Total addressed pages = (address space (eg. 2^64)) / page size

3L pages needed = total pages / entries per page

2L pages needed = num 3L pages / entries per page

1L pages needed = num 2L pages / entries per page

If this is > 1, then the schema is invalid (can only have 1 top level page)

Mapping virtual to physical:

What page is the address on = address / frame size (round down)

An answer of 5.04 means the address is on Page 5 (the 6th page)

Offset = address - (pagenum \* page size)

Look for the page number (5) in the table, and what it maps to (r)  
  
 Physical address = (r \* page size) + offset

Page Faults:

Occur if frame is valid, but not in memory.

(Kernel raises an exception, and puts the page in memory)

Segmentation fault:

Occur if the frame is invalid (no matter what)

Also occur if trying to write to an un-writable page