

Final Exam Review

ICS332 — Operating Systems

Henri Casanova (henric@hawaii.edu)

Spring 2018

What to study

- How comprehensive is it?
- About 30% of the points are about midterm material (i.e., up to and including "Synchronization")
- About 70% is about post-midterm material
- As usual, you should study lecture notes, homework and programming assignments, quizzes
- As usual, there should be very few surprises
- The exam is closed-notes
- No calculator needed

Short-answer questions

- The exam will have a few simple short-answer questions
- When you review the lecture notes, just pretend you have to come up with such questions
- Make sure you review your quizzes
- Let's see a few samples...

Sample short-answer questions

- What is the goal of the TLB?

Sample short-answer questions

- What is the goal of the TLB?
- To speed up address translation

Sample short-answer questions

- What is the goal of the TLB?
- To speed up address translation
- What characteristic of programs makes the TLB be effective?

Sample short-answer questions

- What is the goal of the TLB?
- To speed up address translation
- What characteristic of programs makes the TLB be effective?
- Locality

Sample short-answer questions

- What is the goal of the TLB?
- To speed up address translation
- What characteristic of programs makes the TLB be effective?
- Locality
- Why would we ever want a 2-level hierarchical page table?

Sample short-answer questions

- What is the goal of the TLB?
- To speed up address translation
- What characteristic of programs makes the TLB be effective?
- Locality
- Why would we ever want a 2-level hierarchical page table?
- To avoid having a contiguous page table that's bigger than a page

Sample short-answer questions

- What is the goal of the TLB?
- To speed up address translation
- What characteristic of programs makes the TLB be effective?
- Locality
- Why would we ever want a 2-level hierarchical page table?
- To avoid having a contiguous page table that's bigger than a page
- Why are hard drives slow?

Sample short-answer questions

- What is the goal of the TLB?
- To speed up address translation
- What characteristic of programs makes the TLB be effective?
- Locality
- Why would we ever want a 2-level hierarchical page table?
- To avoid having a contiguous page table that's bigger than a page
- Why are hard drives slow?
- Because of moving parts

Sample short-answer questions

- What is the goal of the TLB?
- To speed up address translation
- What characteristic of programs makes the TLB be effective?
- Locality
- Why would we ever want a 2-level hierarchical page table?
- To avoid having a contiguous page table that's bigger than a page
- Why are hard drives slow?
- Because of moving parts
- What is a problem with SSDs?

Sample short-answer questions

- What is the goal of the TLB?
- To speed up address translation
- What characteristic of programs makes the TLB be effective?
- Locality
- Why would we ever want a 2-level hierarchical page table?
- To avoid having a contiguous page table that's bigger than a page
- Why are hard drives slow?
- Because of moving parts
- What is a problem with SSDs?
- Write amplification

Sample short-answer questions

- What is the goal of the TLB?
- To speed up address translation
- What characteristic of programs makes the TLB be effective?
- Locality
- Why would we ever want a 2-level hierarchical page table?
- To avoid having a contiguous page table that's bigger than a page
- Why are hard drives slow?
- Because of moving parts
- What is a problem with SSDs?
- Write amplification
- BestFit is an algorithm to solve the ... problem

Sample short-answer questions

- What is the goal of the TLB?
- To speed up address translation
- What characteristic of programs makes the TLB be effective?
- Locality
- Why would we ever want a 2-level hierarchical page table?
- To avoid having a contiguous page table that's bigger than a page
- Why are hard drives slow?
- Because of moving parts
- What is a problem with SSDs?
- Write amplification
- BestFit is an algorithm to solve the ... problem
- Contiguous Memory Allocation

Sample short-answer questions

- What is the goal of the TLB?
- To speed up address translation
- What characteristic of programs makes the TLB be effective?
- Locality
- Why would we ever want a 2-level hierarchical page table?
- To avoid having a contiguous page table that's bigger than a page
- Why are hard drives slow?
- Because of moving parts
- What is a problem with SSDs?
- Write amplification
- BestFit is an algorithm to solve the ... problem
- Contiguous Memory Allocation
- Without a TLB, each time I access a memory location I actually access how many memory locations?

Sample short-answer questions

- What is the goal of the TLB?
- To speed up address translation
- What characteristic of programs makes the TLB be effective?
- Locality
- Why would we ever want a 2-level hierarchical page table?
- To avoid having a contiguous page table that's bigger than a page
- Why are hard drives slow?
- Because of moving parts
- What is a problem with SSDs?
- Write amplification
- BestFit is an algorithm to solve the ... problem
- Contiguous Memory Allocation
- Without a TLB, each time I access a memory location I actually access how many memory locations?
- 2

Sample short-answer questions

- What is a problem with Contiguous Memory Allocation?

Sample short-answer questions

- What is a problem with Contiguous Memory Allocation?
- Fragmentation

Sample short-answer questions

- What is a problem with Contiguous Memory Allocation?
- **Fragmentation**
- Does paging remove all fragmentation problems?

Sample short-answer questions

- What is a problem with Contiguous Memory Allocation?
- Fragmentation
- Does paging remove all fragmentation problems?
- No, there is still internal fragmentation

Sample short-answer questions

- What is a problem with Contiguous Memory Allocation?
- **Fragmentation**
- Does paging remove all fragmentation problems?
- **No, there is still internal fragmentation**
- Without a dirty bit, what would be a problem?

Sample short-answer questions

- What is a problem with Contiguous Memory Allocation?
- Fragmentation
- Does paging remove all fragmentation problems?
- No, there is still internal fragmentation
- Without a dirty bit, what would be a problem?
- Useless disk writes

Sample short-answer questions

- What is a problem with Contiguous Memory Allocation?
 - **Fragmentation**
 - Does paging remove all fragmentation problems?
 - **No, there is still internal fragmentation**
 - Without a dirty bit, what would be a problem?
 - **Useless disk writes**
-
- Send me e-mail if you want Quiz #8...

Make sure you know the full "memory narrative"

- What is the complete sequence of steps, from a TLB miss to paging in from disk, for translating a virtual address to a physical address?

Make sure you know the full "memory narrative"

- What is the complete sequence of steps, from a TLB miss to paging in from disk, for translating a virtual address to a physical address?
- Split address into page number and offset (hardware)
- Look up the TLB to find the corresponding frame number (hardware)
- If found in the TLB, build the physical address (hardware) **[DONE]**
- If not found in the TLB, then lookup the page table (hardware)
- If page table entry is valid, then build the physical address (hardware) **[DONE]**
- If not, then issue a trap to the OS with a page fault and puts the process into a blocked state
- The OS sees if there is a free frame in RAM
- If there isn't, the OS selects a victim page
- If the victim is dirty, then the OS writes it back to disk
- The OS marks the frame of the victim as free
- The OS loads the missing page to RAM into the free frame
- The OS updates the process' page table and schedules the process again
- The process issues the logical address (again)
- Split address into page number and offset (hardware)
- Lookup TLB to find the corresponding frame number (hardware)
- It will NOT be found in the TLB, so lookup the page table to find the frame number
- Update the TLB with the page table entry
- Build the physical address

- You can expect exercises similar to what we've seen in Homework Assignments
- If you understood the homework assignments well, there should be any problem

Memory Management in a (large) Nutshell

- What are the mechanisms for implementing MM?
 - Physical and virtual addressing
 - Base and Limit registers
 - Paging, Segmentation
 - Page tables, TLB
 - What are the policies related to MM?
 - Page replacement policy
 - Frame allocation policy
- Virtualizing memory
 - Difference between virtual and physical address
 - Breakdown of virtual addresses into page number, offset
 - Single/Multi-level page tables
 - Valid bit, Dirty bit, Reference bit
 - Inverted page tables

Page table sample (easy) question

- Given 22-bit logical addresses, and a 64KiB page size, how is a logical address split into page number and offset for a single-level page table?

Page table sample (easy) question

- Given 22-bit logical addresses, and a 64KiB page size, how is a logical address split into page number and offset for a single-level page table?
- $64\text{KiB} = 2^{16}$ bytes
- Therefore: offset is 16-bit
- Therefore: page number is $22 - 16 = 6$ -bit

Page table sample (harder) question

- Given a 32-bit address space, and a 8KiB page size, how is the address split into page number and offset for a 2-level page table, **assuming that the outer page table fits exactly in one page?** Assuming 4-byte page table entries, how many bytes are “wasted” due to unused entries?

Page table sample (harder) question

- Given a 32-bit address space, and a 8KiB page size, how is the address split into page number and offset for a 2-level page table, **assuming that the outer page table fits exactly in one page?** Assuming 4-byte page table entries, how many bytes are “wasted” due to unused entries?
- 8KiB = 2^{13} bytes
- Therefore: offset is 13-bit
- Number of page table entries that can fit in a page: $2^{13}/4 = 2^{11}$
- The outer virtual page number is 11-bit (because that outer page table fits exactly in on page)
- Remains $32-13-11 = 8$ bits for the inner virtual page number
- Therefore, each inner page table page contains 2^8 entries, even though it could contain 2^{11} entries
- Therefore only 1/8-th of each inner page table page is used, wasting $\frac{7}{8} \times 2^{11} \times 4 = 7 \times 2^{10}$ bytes
- There are 2^{11} inner page table pages
- So the total waste is $7 \times 2^{10} \times 2^{11} = 7 \times 2^{21}$ bytes = 14 MiB

Another "incomplete" page table question

- Consider a 24-bit address space, and a 8K page size. We know that the single-level page table uses only 1/2 a page. How big are the page table entries?

Another "incomplete" page table question

- Consider a 24-bit address space, and a 8K page size. We know that the single-level page table uses only 1/2 a page. How big are the page table entries?
- 24-bit address space is 2^{24} bytes
- Page size: 2^{13} bytes
- Number of pages: $2^{24}/2^{13} = 2^{11}$
- Number of page table entries: 2^{11}
- Let s be the size of a page table entry
- We have: $2^{11} \times s = (1/2) \times 2^{13}$
- Which gives: $s = 2^{(12-11)} = 2$ bytes

Typical Page Table Question

- Page size = 500 bytes, decimal addresses

vpage #	frame #	valid	dirty
0	3	1	1
3	14	1	
4	12	0	
1	11	1	1
2	33	1	
200		0	

- Was (virtual) address 1900 written to?
- What does (virtual) address 999 translate to?
- Give a virtual address that'll cause a page fault
- What (virtual) address corresponds to byte 7321 in physical memory?
- Is it possible that (virtual) address 132 was written to?

Typical Page Table Question

- Page size = 500 bytes, decimal addresses

vpage #	frame #	valid	dirty
0	3	1	1
3	14	1	
4	12	0	
1	11	1	1
2	33	1	
200		0	

- Was (virtual) address 1900 written to?

Typical Page Table Question

- Page size = 500 bytes, decimal addresses

vpage #	frame #	valid	dirty
0	3	1	1
3	14	1	
4	12	0	
1	11	1	1
2	33	1	
200		0	

- Was (virtual) address 1900 written to?
- Virtual address 1900 is in page $1900/500 = 3$
- The dirty bit for the entry for page 4 is NOT set
- So the answer is: NO

Typical Page Table Question

- Page size = 500 bytes, decimal addresses

vpage #	frame #	valid	dirty
0	3	1	1
3	14	1	
4	12	0	
1	11	1	1
2	33	1	
200		0	

- What does (virtual) address 999 translate to?

Typical Page Table Question

- Page size = 500 bytes, decimal addresses

vpage #	frame #	valid	dirty
0	3	1	1
3	14	1	
4	12	0	
1	11	1	1
2	33	1	
200		0	

- What does (virtual) address 999 translate to?
- Virtual address is in page $999/500 = 1$
- Offset within the page is $999 \% 499 = 499$
- Page 1 is in frame #11
- The physical address is thus $11 * 500 + 499 = 5999$

Typical Page Table Question

- Page size = 500 bytes, decimal addresses

vpag e #	frame #	valid	dirty
0	3	1	1
3	14	1	
4	12	0	
1	11	1	1
2	33	1	
200		0	

- Give a virtual address that'll cause a page fault

Typical Page Table Question

- Page size = 500 bytes, decimal addresses

vpage #	frame #	valid	dirty
0	3	1	1
3	14	1	
4	12	0	
1	11	1	1
2	33	1	
200		0	

- Give a virtual address that'll cause a page fault
- The page table entry for Page #4 is marked as invalid, so let's access it
- For instance, $4 \times 500 + 42$ is in page 4
- Therefore, accessing address 2042 will cause a page fault

Typical Page Table Question

- Page size = 500 bytes, decimal addresses

vpage #	frame #	valid	dirty
0	3	1	1
3	14	1	
4	12	0	
1	11	1	1
2	33	1	
200		0	

- What (virtual) address corresponds to byte 7321 in physical memory?

Typical Page Table Question

- Page size = 500 bytes, decimal addresses

vpage #	frame #	valid	dirty
0	3	1	1
3	14	1	
4	12	0	
1	11	1	1
2	33	1	
200		0	

- What (virtual) address corresponds to byte 7321 in physical memory?
- This address is in frame $7321/500 = 14$
- Per the page table, frame 4 contains page 3
- The offset in the frame is $7321\%500 = 321$
- Therefore, the virtual address is $3 * 500 + 321 = 1821$

Typical Page Table Question

- Page size = 500 bytes, decimal addresses

vpage #	frame #	valid	dirty
0	3	1	1
3	14	1	
4	12	0	
1	11	1	1
2	33	1	
200		0	

- Is it possible that virtual address 132 was written to?

Typical Page Table Question

- Page size = 500 bytes, decimal addresses

vpage #	frame #	valid	dirty
0	3	1	1
3	14	1	
4	12	0	
1	11	1	1
2	33	1	
200		0	

- Is it possible that virtual address 132 was written to?
- Virtual address 132 is in page 0
- The entry for page 0 says that the page is dirty
- So yes, it's possible

Typical Page Replacement Question

- Given a number of frames, given a sequence of logical page references, and given a page replacement algorithm, determine which page references will page fault
 - We've seen only 3 algorithms: FIFO, LRU, Optimal
 - FIFO and LRU you implemented in Homework #8
- Let's briefly flip through the one example we say in class..

Optimal Page Replacement

references	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
frame #0	7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7	7	7
frame #1		0	0	0	0	0	0	4	4	3	3	3	3	1	1	1	1	1	1	1
frame #2			1	1	1	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0
page faults	X	X	X	X		X		X		X				X				X		

FIFO Page Replacement

references	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
frame #0	7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7	7
frame #1		0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0	0
frame #2			1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2	1
page faults	X	X	X	X		X	X	X	X	X	X			X	X			X	X	X

LRU Page Replacement

reference	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
frame #0	7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	1
frame #1		0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0
frame #2			1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	7	7	7
page faults	X	X	X	X		X		X	X	X	X			X		X		X		

Swapping

- Make sure you're ready to answer questions about Swapping
 - Why does it occur?
 - What are solutions?
- **Sample Question:** Consider a server that's currently running many processes, none of them doing a lot of I/O, and yet we observe 20% CPU utilization and 99.9% disk utilization. Which of these options would help this situation:
 - Install a faster CPU.
 - Install a bigger disk.
 - Allow more processes into the ready queue
 - Kill some processes
 - Buy more RAM
 - By a faster disk

The End

- Any question about pre-midterm material?
- Any homework assignment solution we should go through?
 - If time, let's look briefly at my solution for Homework Assignment #8...
- Any random questions?