

# CMPU 334 Quiz1

Fall 2021

Name: \_\_\_\_\_

## Instructions:

This is a closed book, closed notes exam. No electronic devices, including calculators, are allowed. You have 75 minutes. There are 7 problems and 11 pages to this exam.

Good Luck!

1 (15)	
2 (13)	
3 (17)	
4 (15)	
5 (12)	
6 (10)	
7 (18)	
Total (100)	

### 1. (15 points) Limited Direct Execution

Order the steps(1 to 6) in the execution of a system call, with 1 being the first step and 6 being the last step.

\_3\_ jump to trap handler

\_1\_ trap into the OS

\_6\_ return to process

\_5\_ move to user mode

\_2\_ move into kernel mode

\_4\_ return from trap handler

What role does the hardware timer play in helping to provide a Limited Direct Execution environment for processes?

**The timer causes a periodic interrupt which traps into the kernel, allowing it to regain control from a running process.**

## 2. (13 points) Processes

1. Say we have a system with the following three process states:

- Running
- Ready
- Blocked

Which of the following are valid process transitions? **Circle all that apply.**

- A. BLOCKED to RUNNING
- B. BLOCKED to READY**
- C. RUNNING to READY**
- D. RUNNING to BLOCKED**
- E. READY to RUNNING**
- F. READY to BLOCKED

2. Which of the following would be considered the state of a process? **Circle all that apply.**

- A. Process Memory**
- B. Register Values**
- C. State of the TLB
- D. Open files**

3. In Linux, a process that has finished running but the parent hasn't called wait on the child is called what? **Circle One.**

- A. An orphan process
- B. A zombie process**
- C. A blocked process
- D. An init process

### 3. (17 points) Scheduling

1. If you were most concerned about **fairness**, which scheduler would you pick? **Circle One.**
  - A. First In, First Out
  - B. Shortest Job First
  - C. Shortest Time-to-Completion First
  - D. Round Robin**
  
2. If you were most concerned about **turnaround time**, which scheduler would you pick? **Circle One.**
  - A. First In, First Out
  - B. Shortest Time-to-Completion First**
  - C. Round Robin
  - D. Linux CFS scheduler
  
3. If you were most concerned about **response time**, which scheduler would you pick? **Circle One.**
  - A. First In, First Out
  - B. Shortest Job First
  - C. Shortest Time-to-Completion First
  - D. Round Robin**
  
4. In a Multi-Level Feedback Queue, adding a priority boost solves which of the following problems? **Circle all that apply.**
  - A. Prevention of gaming the system by requesting I/O at the end of your time slice
  - B. Potential starvation of processes**
  - C. Prioritizing a CPU-bound job that has now become interactive**
  - D. Letting a CPU-bound job have a longer time quantum
  
5. In the Linux CFS scheduler, which of the following statements are **true**? **Circle all that apply.**
  - A. It is always fair for processes over the target scheduling latency.
  - B. Users can control the priority of a process to give it a higher share of the CPU**
  - C. To speed up scheduling, CFS keeps track of only running or runnable processes in a linked list.
  - D. CFS decides which process to run next based on its virtual runtime (vruntime).**

#### 4. (15 points) Memory

1. Which of the following statements about `malloc()` are true? **Circle all that apply.**

- A. `malloc` returns -1 if it can not fulfill the request
- B. `malloc` takes a single argument specifying the number of bytes of requested memory**
- C. `malloc` returns the actual number of bytes you were allocated
- D. `malloc` returns a pointer to the start of the memory that was allocated**

2. Which of the following statements about `free()` are true? **Circle all that apply.**

- A. `free` takes a pointer previously returned by `malloc`**
- B. `free` will return an error if you free the same pointer twice
- C. Using `free` is recommended but not strictly necessary because C will garbage collect any memory that is no longer in use
- D. When a process exits, allocated memory will be reclaimed by the OS even if `free` was not called**

3. Which of the following statements about segments are true? **Circle all that apply.**

- A. Bounds checking can only be done in the the virtual address space
- B. Bounds checking can only be done in the the physical address space
- C. Bounds checking can be done in either the virtual or physical address space**
- D. A system that uses segments can suffer from external fragmentation**

4. Which of the following statements about the binary buddy allocator are true? **Circle all that apply.**

- A. Requests can suffer from internal fragmentation**
- B. Coalescing free blocks is simple**
- C. It takes  $O(n)$  time to find a free block

## 5. (12 points) Odds and Ends

1. Which of the following are **policies**. **Circle all that apply.**

- A. Walking a page table to find a PFN
- B. Finding a free block with First Fit**
- C. Reducing the priority of a process after it has used up its time quanta**
- D. Limited direct execution

2. Which of the following help to speed up address translation. **Circle all that apply.**

- A. TLB**
- B. Having a multi-level page table
- C. Having base and bounds registers for segments**
- D. Marking the stack as non-executable

3. Which of the following statements about the clock algorithm are **true**? **Circle all that apply.**

- A. Needs hardware support to be efficiently implemented**
- B. Approximates an LRU algorithm**
- C. Used in the Linux CFS scheduler
- D. May not terminate with certain memory access patterns

4. Which of the following statements about the VAX/VMS virtual memory system are **true**? **Circle all that apply.**

- A. It is an example of a hybrid system**
- B. Used bounds registers because of the very large page size used in the system
- C. Utilized the lazy optimization copy on write (COW)**
- D. Had only one segment to facilitate data transfer between user space and the kernel

## 6. (10 points) Page Tables

Assume the following: a 22-bit virtual address space, with a 1024 byte (1 KB) page size.

(a) How many bits are in the **offset** portion of the virtual address?

**10 bits to represent 1024 bytes**

(b) How many bits are in the **VPN** portion of the virtual address?

**12 bits**

Now, let's focus on the page table. Assume each *page table entry* is 4 bytes in size. Assuming a *linear* page table:

(c) How many page table entries (PTEs) are there in the page table?

**You need one entry for each VPN, so  $2^{12} = 4096$  PTEs.**

(d) How many PTEs fit onto a single page in this system?

**$2^{10} / 2^2 = 2^8 = 256$  PTEs per page**

(e) How many **pages** of memory do we need to hold the entire page table?

**$2^{12} / 2^8 = 2^4 = 16$**

## 7. (18 points) Address Translation

The following problem concerns the way virtual addresses are translated into physical addresses.

- The memory is byte addressable.
- Virtual addresses are 16 bits wide.
- Physical addresses are 14 bits wide.
- The page size is 512 bytes.
- The TLB is 4-way set associative with 16 total entries.

In the following tables, **all numbers are given in hexadecimal**. The contents of the TLB and the page table for the first 32 pages are as follows:

TLB			
Index	Tag	PFN	Valid
0	14	08	1
	1F	1F	1
	1A	0A	0
	19	09	1
1	18	1E	0
	17	13	1
	02	01	0
	11	18	0
2	06	17	0
	1C	02	1
	1A	05	1
	1D	16	1
3	1F	14	0
	0E	0B	0
	18	14	1
	10	10	0

**Page Table (first 32 pages only)**

<b>VPN</b>	<b>PFN</b>	<b>Valid</b>	<b>VPN</b>	<b>PFN</b>	<b>Valid</b>
00	1B	1	10	16	0
01	06	0	11	17	0
02	03	1	12	16	1
03	1F	0	13	03	0
04	00	1	14	1D	0
05	13	0	15	1B	0
06	1E	1	16	09	0
07	11	1	17	06	0
08	1A	0	18	0C	1
09	01	1	19	14	1
0A	08	1	1A	0F	0
0B	05	1	1B	12	1
0C	16	1	1C	10	0
0D	1F	0	1D	17	1
0E	1D	0	1E	19	1
0F	10	0	1F	0A	1

For the given virtual address, indicate the TLB entry accessed and the physical address. Indicate whether the TLB misses and whether a page fault occurs.

If there is a page fault, enter "N/A" for "PFN" and leave part (c) blank.

**Virtual address** : 0x15A8

(a) Virtual address format (one bit per box)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	0	1	0	1	1	0	1	0	1	0	0	0

(b) Address translation

Parameter	Value
VPN	0x0A
TLB Index	0x2
TLB Tag	0x02
TLB Hit? (Y/N)	N
Page Fault? (Y/N)	N
PFN	0x8

(c) Physical address format (one bit per box)

13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	1	0	1	0	1	0	0	0

Physical address: 0x11A8 \_\_\_\_\_

For the given virtual address, indicate the TLB entry accessed and the physical address. Indicate whether the TLB misses and whether a page fault occurs.

If there is a page fault, enter “N/A” for “PFN” and leave part (c) blank.

**Virtual address** : 0xBA93

(b) Virtual address format (one bit per box)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	1	0	1	0	1	0	0	1	0	0	1	1

(c) Address translation

Parameter	Value
VPN	0x5D
TLB Index	0x1
TLB Tag	0x17
TLB Hit? (Y/N)	Y
Page Fault? (Y/N)	N
PFN	0x13

(d) Physical address format (one bit per box)

13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	1	1	0	1	0	0	1	0	0	1	1

Physical address: 0x2693\_\_\_\_\_