CS 537: Introduction to Operating Systems
Spring 2024: Midterm Exam #1

This exam is closed book, closed notes.

No calculators may be used.

You have 1 hour and 30 minutes to complete this exam.

Write all of your answers on the accu-scan form with a #2 pencil:

- CANVAS LAST NAME - fill in your last (family) name starting at the leftmost column
- CANVAS FIRST NAME - fill in the first five letters of your first (given) name
- IDENTIFICATION NUMBER - This is your UW Student WisCard ID number
- ABC of SPECIAL CODES - Write your lecture number as a three digit value 001 or 002.

These exam questions must be returned at the end of the exam, but we will not grade anything in this booklet.

You may separate the pages of this exam if it helps you.

Unless stated (or implied) otherwise, you should make the following assumptions:

- The OS manages a single uniprocessor (single core)
- All memory is byte addressable
- Page table entries require 4 bytes
- Data is allocated with optimal alignment, starting at the beginning of a page
- Assume leading zeros can be removed from numbers (e.g., 0x06 == 0x6).
- Hex numbers are represented with a proceeding “0x”

The following might help you with some calculations:

- 0x100 = 2^8 = 256
- 2^{10} = 1024
- 2^{12} = 4096
- 2^{10} bytes = 1KB
- 2^{20} bytes = 1MB

This exam has 60 questions. Each question has the same number of points.

Good luck!

Designate if the statement is True (A) or False (B).

1. The address space of a process is part of its process state.
True. The page table contains the translation from address space to physical space.
2. A process is identical to a program.
False, a program is static but a process is dynamic.
3. Two processes reading from the same virtual address will access the same content.
False, each process has its own address space.
4. When an I/O operation completes, the previously blocked process moves into the RUNNING state.
False, A previously running process moves from the BLOCKED state to the RUNNABLE (READY) state. The scheduler decides which process moves to the RUNNING state from those that are RUNNABLE.
5. The fork() system call clones the calling process and executes a different program in the child process.
False, The fork() system call clones the calling process but it does NOT start a new program in the child process. That is the exec() system call.
6. A modern OS virtualizes memory by time-sharing.
False, memory is virtualized by space-sharing.
7. Hardware support of a timer interrupt is required for cooperative multi-tasking.
False, cooperative assumes processes will give up the CPU voluntarily.
8. An SJF scheduler can suffer from the convoy effect where a lot of short-running jobs can get stuck behind a long-running job.
True, Jobs that arrive after a long job has already been scheduled would need to wait, since SJF is non-preemptive.
9. A RR scheduler may preempt the currently running job.
True, RR is a preemptive scheduler.
10. An STCF scheduler cannot cause jobs to starve.
False, if a constant stream of short-running jobs arrive then a long-running job will constantly be put off and never scheduled (i.e. starve).
11. If all jobs arrive at the same time, an SJF and an STCF scheduler will behave the same.
True, since all jobs arrive at the same time, STCF will behave like SJF.
12. With Limited Direct Execution if a user program attempts to perform a privileged operation then a system call is executed.
False, if a user attempts to perform a privileged instruction an interrupt is raised and a trap handler executes to shut the program down.
13. A return-from-trap instruction is used at the end of a system call to reduce the privilege level and jump back to the calling process's code.
True, that is what happens at the end of a system call, in addition to restoring the process's state.
14. With an MLFQ scheduler, jobs run to completion (i.e. are not preempted) as long as there is not a higher priority job.
False, multiple jobs at the same priority level will be scheduled with RR.
15. With copy-on-write paging, a page is marked read-only if the reference count = 1.
False, a page is marked as read-only if there are multiple references to the page. If a process tries to write to the page, a page fault occurs and the OS realizes that it's a COW page and thus lazily allocates a new page for the process.
16. The return-value of fork() is 0 in the child process and the PID of the child process in the parent process.
True, look at the man page.
17. A context switch occurs when a process performs a system call.
False, A context switch occurs when one process is preempted and a different process is scheduled.

18. The STCF scheduler minimizes average turnaround time.
True, STCF minimizes turnaround time.
19. One purpose of the TLB is to reduce page table size.
False, the TLB reduces the number of memory accesses.
20. The valid bit in the TLB signifies if the page is in memory or on disk.
False, The TLB does not store the present bit, the page table entry stores that. The valid bit in the TLB signifies if a cache entry has a translation or should be ignored.

Select the one best answer, A - E.

Process States

Assume you have a system with three processes (A, B, and C) and a single CPU. Assume an MLFQ scheduler. Processes can be in one of five states: RUNNING, READY, BLOCKED, not yet created, or terminated. Given the following cumulative timeline of process behavior, indicate the state the specified process is in AFTER that step, and all preceding steps, have taken place.

For all questions in this part, use the following options for each answer:

A. RUNNING
B. READY
C. BLOCKED
D. Process has not been created yet
E. Process has been terminated

Step 1: Process A is loaded into memory and begins; it is the only user-level process in the system.

21. Process A is in what state?
A, RUNNING
Step 2: Process A calls fork() and creates process B. Process A is scheduled.

22. Process A is in what state?
A, RUNNING
23. Process B is in what state?
B, READY
Step 3: The running process issues an I/O request to the disk.

24. Process A is in what state?
C, BLOCKED since it issued an I/O request
25. Process B is in what state?
A, RUNNING since it was the only READY process
Step 4: The running process calls fork() and creates process C. Process C is not yet scheduled.

26. Process A is in what state?
C, BLOCKED (no change)
27. Process B is in what state?
A, RUNNING (no change; C is not scheduled so it can't be running)
28. Process C is in what state?
B, READY
Step 5: The time-slice of the running process expires. Process C is scheduled.

29. Process A is in what state?
C, BLOCKED (no change)

30. Process B is in what state?
B, READY (time-slice expired, but it hasn't finished yet)
31. Process C is in what state?
A, RUNNING (it was scheduled)
Step 6: The previously issued I/O request completes; the process that issued the I/O request is scheduled.

32. Process A is in what state?
A, RUNNING (I/O completed, now scheduled)
33. Process B is in what state?
B, READY (no change)
34. Process C is in what state?
B, READY (something else was scheduled)
CPU Job Scheduling

Assume a workload with the following characteristics. If needed, assume a time-slice of 1 second and if there is a choice the newest arriving job is selected.

Job Name	Arrival Time (seconds)	CPU Burst Time (seconds)
A	0	9
B	2	3
C	6	5

35. Given a FIFO scheduler, what is the turnaround time of job B?
A. 5
B. 9
C. 10
D. 11
E. None of the above
C, 10 seconds. Time Slices: AAAAAAAAABBBCCCCC, 12 - 2 = 10
36. Given a FIFO scheduler, what is the average turnaround time of the three jobs?
A. 8
B. 9
C. 10
D. 11
E. None of the above
C, 10 seconds. A is 9 - 0 = 9, B is 12 - 2 = 10, C is 17 -6 = 11
37. Given a SJF scheduler, what is the turnaround time of job C
A. 5
B. 9
C. 10
D. 11
E. None of the above
D, 11 seconds. Same as FIFO since SJF has no preemption.

38. Given a RR scheduler, what is the turnaround time of job B
A. 3
B. 5
C. 7
D. 9
E. None of the above
E, 6 seconds. 8 - 2 = 6. Time slices are AABABACBACACACACA. Note that at time 2, B is the newest job so it runs. At time 6, C arrives and it is the newest job so it runs first and this is why B only runs afterward.
39. Given a STCF scheduler, what is the average turnaround time of the three jobs?
A. 8.33
B. 9
C. 10
D. 11
E. None of the above
A, 8.33 seconds. A is 17 - 0, B is 5 - 2 = 3, C is 11 - 6 = 5. Time Slices are AABBBACCCCCAAAAAA.
40. Give a FIFO scheduler, what is the response time of job B
A. 3
B. 5
C. 7
D. 9
E. None of the above
C, 7 seconds. First runs at time slice 9. Arrival time is 2. 9 - 2 = 7
41. Assume a new job D arrives at some time T, requiring 4 seconds of CPU time. For which arrival times, T, would D preempt the job running at that time with the STCF scheduler?
A. 3
B. 8
C. 9
D. 14
E. None of the above
E, None of the above. At each of the above times, the running process has less than 4 seconds left.

Address Translation

Assume you have an architecture with a 1 KB address space and 16 KB of physical memory. You are performing dynamic address translation using **segmentation with base and bounds registers**. The top bit of a virtual address specifies the segment. Translate each of the following virtual addresses to their physical address. All segments grow in the positive direction.

Segment 0: Base 0x300c (decimal 12300) Bounds 392 (decimal)

Segment 1: Base 0x1ef2 (decimal 7922) Bounds 450 (decimal)

42. Virtual Address 0x36b (decimal 875)
A. 0x3177 (decimal 12663)
B. 0x205d (decimal 8285)
C. 0x136b (decimal 4971)
D. Segmentation Violation

E. None of the above
B. Top bit is a 1 so segment 1. Decimal value of the rest of the VA is 363 (less than the bounds of 450). 0x16b + 0x1ef2 = 0x205d
43. Virtual Address 0x3c4 (decimal 964)
A. 0x20b6 (decimal 8299)
B. 0x31d0 (decimal 12752)
C. 0x33d0 (decimal 13264)
D. Segmentation Violation
E. None of the above
D. Top bit is a 1 so segment 1. Decimal value of the rest of the VA is 452 (greater than the bounds of 450).
44. Virtual Address 0x17d (decimal 381)
A. 0x206f (decimal 8303)
B. 0x318a (decimal 12682)
C. 0x1227 (decimal 4647)
D. Segmentation Violation
E. None of the above
E. Top bit is a 0 so segment 0. Decimal value of the rest of the VA is still 381. 381 + 12300 = 12681 (0x3189)
On a new system dynamic relocation is performed with **a linear page table**. Assume a system with the following parameters:

Address Space size is 32 KB
Physical Memory size is 128 KB
Page size is 4KB

You are given the following trace of virtual addresses and the physical addresses they translate to. Can you reverse engineer the contents of the page table for this process?
VA 0x00006e19 --> 0x0003e19
VA 0x00004d35 --> 0x000ad35
VA 0x000030d8 --> 0x00050d8
VA 0x0000244d --> 0x001a44d
VA 0x00005665 --> Invalid

In VA 0x006e19 the underlined 6 is the VPN (based on page size 4KB -> 12 offset bits)
From the trace we see VPN 6 -> PFN 3, 4 -> 0xa, 3 -> 5, 2 -> 0x1a

45. Page Table Entry 0
A. Valid, PFN = 0x00
B. Valid, PFN = 0xd0
C. Invalid Entry or page table does not contain entry for this VPN
D. Contents of PTE cannot be determined from this trace
E. None of the above
D, no uses of VPN 0 in trace.
46. Page Table Entry 2
A. Valid, PFN = 0x01
B. Valid, PFN = 0x1a
C. Invalid Entry or page table does not contain entry for this VPN
D. Contents of PTE cannot be determined from this trace
E. None of the above
B, VPN 2 used and translated to 0x1a PFN

47. Page Table Entry 3
A. Valid, PFN = 0x06
B. Valid, PFN = 0x63
C. Invalid Entry or page table does not contain entry for this VPN
D. Contents of PTE cannot be determined from this trace
E. None of the above
E, none of the above. VPN 3 used and translated to 0x5 PFN
48. Page Table Entry 5
A. Valid, PFN = 0x03
B. Valid, PFN = 0x30
C. Invalid Entry or page table does not contain entry for this VPN
D. Contents of PTE cannot be determined from this trace
E. None of the above
C, invalid entry. VPN 5 used and resulted in invalid.
49. Page Table Entry 8
A. Valid, PFN = 0x03
B. Valid, PFN = 0x1a
C. Invalid Entry or page table does not contain entry for this VPN
D. Contents of PTE cannot be determined from this trace
E. None of the above
C, only 8 virtual pages index 0 through 7.
A different system uses a **2-level page table** with the following parameters:

Page Sizes are 32 bytes
Virtual Address space is 1024 pages (or 32 KB)
Physical Memory consists of 128 pages
A Virtual Address needs 15 bits (5 for the offset, 10 for the VPN)
A Physical address needs 12 bits (5 for the offset, 7 for the PFN)

The system uses a multi-level page table. The upper five bits of a VA are the index into the page directory to get the PDE. If the PDE is valid, it points to a page of the page table. Each page holds 32 PTEs. If the PTE is valid, it holds the desired translation (PFN).

The format of a PTE is:

VALID | PFN6 ... PFN0

The format of a PDE is identical:

VALID | PT6 ... PT0

The PDBR holds the value: 0x7e (decimal: 126) [This means the page directory is held in this page]

The content of physical memory is on the next two pages. The left-most column shows the

physical page number in hex and decimal. The top two rows show the offset of the bytes in the pages in both hex and decimal.

```
hex offset   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
dec offset   0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
0x00 (  0): 00 09 1d 19 03 09 0f 19 0f 16 16 02 15 1a 0c 1c 08 18 10 04 01 06 09 0e 1a 0a 1b 01 07 14 0b 1e
0x01 (  1): 01 06 0f 14 16 11 16 0a 03 14 19 04 08 12 09 0d 0e 10 1e 02 10 15 0d 19 10 13 15 01 07 04 1d 12
0x02 (  2): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x03 (  3): 14 10 1b 0b 0e 14 11 1c 04 0b 05 07 09 18 1a 0c 01 16 0a 04 06 1a 10 09 0d 16 13 14 1a 02 05 07
0x04 (  4): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x05 (  5): 1a 15 1e 1d 00 15 07 1e 15 18 02 1e 0d 1d 01 1c 0c 17 04 04 07 0e 0b 12 1d 06 08 00 0a 0d 18 05
0x06 (  6): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x07 (  7): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f cb 7f aa 7f 7f 7f dc 7f 7f
0x08 (  8): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f af ab 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f f2 7f 7f
0x09 (  9): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0a ( 10): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0b ( 11): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c ( 12): 7f 7f 7f 7f a7 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f c5 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f d3 7f
0x0d ( 13): 18 0c 01 09 1c 14 0f 12 04 06 18 17 10 07 18 0c 19 1e 07 1c 0f 11 13 08 0e 00 1b 05 11 12 00 12
0x0e ( 14): 7f 7f 7f 7f db 7f 7f f4 7f df 7f 7f 7f 7f 7f 7f 7f 7f 98 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
0x0f ( 15): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10 ( 16): 7f 7f 7f 7f 7f 7f 7f ac 7f 7f 7f 7f a2 7f 7f 7f 7f 7f 7f 7f ed 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
0x11 ( 17): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x12 ( 18): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x13 ( 19): 0c 10 13 11 14 1b 05 0e 09 04 0f 16 1d 11 16 03 15 00 03 0f 0f 09 07 00 12 18 17 04 11 1e 1b 07
0x14 ( 20): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f e0 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f a8 7f 7f 7f 7f
0x15 ( 21): 1e 10 00 05 10 18 10 03 1a 0c 0a 13 0f 11 19 1e 08 0e 09 15 1d 0f 04 10 16 11 01 01 1e 13 06 0f
0x16 ( 22): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f e1 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
0x17 ( 23): 0d 15 19 1e 12 17 00 17 14 16 09 15 0d 00 09 0f 03 04 0f 08 15 16 0a 11 09 18 19 13 07 1e 0d 03
0x18 ( 24): 16 0f 03 10 14 0f 1d 1d 19 1c 10 07 06 16 13 17 1d 13 12 0f 1a 07 13 18 1e 0b 17 03 1e 04 05 0a
0x19 ( 25): 7f 7f 7f 7f 7f d1 7f 7f b1 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 85 7f 7f 7f 7f f3 7f 7f 7f 7f 7f 7f 7f
0x1a ( 26): 0e 13 01 1a 01 07 0c 15 03 1d 08 1e 18 1e 05 02 06 14 0f 12 0c 00 10 19 11 0f 17 02 08 15 02 14
0x1b ( 27): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1c ( 28): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1d ( 29): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1e ( 30): 01 01 13 11 1c 17 11 04 03 1c 08 0f 02 04 06 0f 08 0f 0e 0b 11 12 12 19 10 14 0d 08 0c 06 02 0a
0x1f ( 31): 0a 1c 18 15 06 13 1c 1b 11 09 09 07 06 0d 03 09 1d 13 1e 18 15 1e 0d 05 00 0f 0a 0f 13 01 0f 1e
0x20 ( 32): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x21 ( 33): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x22 ( 34): 06 13 13 11 0f 10 07 04 1d 0d 03 17 08 1a 0b 09 1b 08 01 0b 16 0d 00 11 0e 08 02 09 18 04 10 15
0x23 ( 35): 0a 15 0d 11 0b 06 18 01 09 05 11 0e 16 1c 07 02 04 05 06 04 0c 0c 0e 1b 0a 04 16 19 12 02 17 0e
0x24 ( 36): 15 1c 13 1b 1d 1d 0a 0b 14 02 02 12 06 15 0b 12 1d 11 17 05 1a 0a 01 10 12 17 0a 19 01 02 13 17
0x25 ( 37): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x26 ( 38): 0e 05 16 06 0f 1a 0e 11 01 08 15 09 1d 0b 00 0b 07 13 11 0a 1a 09 07 03 1a 1a 05 14 0c 09 0d 08
0x27 ( 39): 03 13 13 18 1d 19 17 00 10 1a 04 16 03 18 09 03 0f 06 18 1d 09 13 03 1d 09 02 1d 13 00 0b 1e 1c
0x28 ( 40): 1b 06 0d 11 11 12 1e 0f 04 13 0b 03 15 03 01 1c 16 07 11 1c 1d 15 13 02 02 05 16 1b 0c 1c 18 06
0x29 ( 41): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x2a ( 42): 04 0c 08 1b 14 08 19 1c 10 06 09 05 1b 06 1d 04 0a 17 05 1b 04 10 1a 1e 1a 13 14 04 02 0f 17 11
0x2b ( 43): 01 02 15 15 1a 01 05 0f 1b 0b 05 0c 03 08 00 0c 19 1a 0f 03 14 16 0a 15 0d 1b 10 0f 01 10 0c 05
0x2c ( 44): 0f 08 0b 1b 1d 0c 13 1c 18 1c 16 09 10 00 11 18 14 11 01 0b 01 04 1a 17 07 06 17 14 11 17 0c 17
0x2d ( 45): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x2e ( 46): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 8d 7f 7f
0x2f ( 47): 14 09 01 17 1e 11 04 0f 0f 1c 17 01 02 05 13 15 01 1b 02 1e 0c 03 12 06 08 06 07 1b 19 1b 03 09
0x30 ( 48): 11 14 03 10 1d 02 01 0d 17 1c 1e 02 0d 10 08 06 14 01 0b 08 1c 19 0b 10 0c 14 0b 13 04 16 12 15
0x31 ( 49): 07 1c 03 02 15 01 00 18 0a 11 11 19 0b 13 04 1c 0e 0a 03 17 19 18 10 11 11 0e 17 0e 19 1d 14 08
0x32 ( 50): 03 08 15 11 16 0c 13 0c 03 0e 05 0e 06 0a 08 01 09 1b 15 12 18 1d 0f 11 0f 10 1b 12 1c 1c 0d 09
0x33 ( 51): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x34 ( 52): 0d 06 05 11 00 1e 14 11 00 0a 0f 09 1a 0d 08 1d 02 03 11 18 14 17 00 0b 0b 07 10 0b 03 0b 13 05
0x35 ( 53): 7f 7f 7f 7f 7f 7f 7f b2 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f a6 7f 7f 7f 7f 7f 7f 7f
0x36 ( 54): 14 1a 0c 0a 13 00 1d 1d 1c 06 04 16 03 15 04 13 02 03 02 11 16 1a 16 18 16 0d 1b 0c 0a 16 0a 0a
0x37 ( 55): 7f 7f 7f 7f 9a 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f f9 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 95 7f 7f 7f 7f
0x38 ( 56): 18 0a 10 1b 01 03 1b 11 07 13 06 17 07 13 14 1a 14 00 1e 0d 1d 07 0b 18 17 04 00 11 1e 03 0e 09
0x39 ( 57): 11 18 11 09 05 1a 0c 1e 12 05 15 01 13 14 0c 1c 16 17 1b 09 17 09 15 14 16 1d 15 13 19 18 0d 19
0x3a ( 58): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x3b ( 59): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f b4 7f 7f 7f 7f b0 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
0x3c ( 60): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x3d ( 61): 1e 02 08 10 0a 13 08 02 17 05 0c 0c 0d 0f 02 02 05 0e 16 10 1a 15 0a 11 11 12 14 1c 1b 12 0d 1c
0x3e ( 62): 15 14 05 07 12 08 13 0b 16 01 17 15 00 02 15 03 08 02 06 14 08 00 13 1a 1d 1d 1c 04 01 13 16 08
```

```
0x3f ( 63): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 97 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 80
0x40 ( 64): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x41 ( 65): 1d 0e 04 05 07 0b 1a 1c 0d 06 02 16 1a 19 03 12 15 1b 17 07 0b 19 16 02 0a 15 1d 0e 01 08 06 19
0x42 ( 66): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x43 ( 67): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x44 ( 68): 16 0c 0b 05 0e 08 11 07 1c 05 04 13 1c 0b 1e 0f 18 18 17 19 13 1d 11 17 13 1e 12 1c 01 13 15 1b
0x45 ( 69): 1a 0a 09 1e 07 13 0e 1b 05 03 12 1e 02 19 07 12 09 12 02 15 07 09 0d 1a 13 07 01 04 05 13 17 14
0x46 ( 70): 0e 0d 12 12 0d 17 02 13 06 17 1e 01 07 01 1c 00 17 1b 04 08 1d 1c 01 1b 05 06 1d 04 14 0e 10 06
0x47 ( 71): 7f 7f 7f 7f 7f 7f 83 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f eb 7f 7f 7f 7f 7f 7f 7f
0x48 ( 72): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x49 ( 73): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f fb 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f d6 7f 7f 7f 7f
0x4a ( 74): 14 06 0c 19 11 14 0b 15 05 00 11 11 14 05 19 07 0f 0b 15 18 1e 18 15 17 08 12 1d 0b 1b 09 15 0a
0x4b ( 75): 13 16 01 02 09 0c 1d 1c 16 1e 01 0d 11 03 1e 1a 16 02 0f 0e 00 04 0d 0b 15 03 1c 12 19 19 0e 12
0x4c ( 76): 16 10 19 0f 13 07 1b 1c 04 18 1e 16 03 0b 05 15 13 1a 00 13 1b 00 0f 03 09 17 16 16 1e 09 00 15
0x4d ( 77): 1d 08 02 0f 04 07 1d 10 15 0f 14 09 1d 01 09 14 12 03 0e 18 11 13 0c 03 0c 05 0e 14 12 04 0c 14
0x4e ( 78): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f fd 7f 7f 7f 7f 7f 7f 7f
0x4f ( 79): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f be 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
0x50 ( 80): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x51 ( 81): 00 0b 12 1b 10 09 08 0d 1a 02 0c 0f 19 0f 03 15 0f 17 12 0e 02 14 08 0c 1e 11 0c 1d 0c 0e 1e 17
0x52 ( 82): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f b6 7f 7f 7f 7f 7f 7f
0x53 ( 83): 1a 02 1c 1b 0d 0d 0a 1c 08 01 0b 16 02 05 04 00 11 15 17 0b 1a 1c 05 11 10 0b 07 10 13 0f 12 1b
0x54 ( 84): 7f 7f 7f 7f 7f c6 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f c1 7f 7f 7f a3 7f 7f 7f 7f 7f 7f 7f
0x55 ( 85): 18 0c 11 04 10 0c 17 16 19 12 1d 15 01 0d 18 15 03 0f 0c 1a 1c 0a 16 13 01 0a 0c 15 09 0b 04 1c
0x56 ( 86): 04 09 0c 17 19 0c 13 01 1e 07 0d 18 01 14 06 17 12 0a 16 18 05 0a 1b 14 0b 07 0c 0f 08 05 03 00
0x57 ( 87): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x58 ( 88): 07 14 0d 11 12 03 10 11 00 06 17 14 14 17 10 11 00 1a 0e 10 1c 0d 10 17 1e 00 1a 0b 12 1b 03 07
0x59 ( 89): 14 19 0c 0b 18 07 07 1d 0c 07 15 0b 0c 09 0b 08 12 11 08 0d 15 06 0b 03 1c 0b 14 16 09 1b 19 1e
0x5a ( 90): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x5b ( 91): 19 0c 04 0d 09 01 18 18 02 08 13 15 1b 04 1e 05 1c 1e 04 14 18 0c 00 11 0f 1a 01 12 19 0c 19 0d
0x5c ( 92): 03 1e 13 0a 04 18 0d 0e 00 16 06 1e 00 04 04 05 03 06 10 01 14 0b 09 01 18 00 0d 03 1a 0e 08 0b
0x5d ( 93): 1b 15 09 18 1a 1d 07 12 00 04 14 1e 01 00 01 00 11 0c 14 09 08 1e 12 09 00 0e 13 06 19 1b 0b 0c
0x5e ( 94): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f cc 7f e5
0x5f ( 95): 0c 1e 0e 03 03 11 02 0a 1c 1c 03 1d 11 17 02 13 0f 13 03 0e 0a 00 14 01 0a 0b 07 19 13 00 0f 00
0x60 ( 96): 0f 1e 19 0a 13 12 11 18 1c 07 06 06 08 04 17 1d 03 0f 14 17 19 1e 14 03 06 19 1c 1e 03 0f 19 12
0x61 ( 97): 17 1d 0a 10 1a 08 07 0a 05 14 09 0f 0b 0e 19 16 14 04 00 1b 08 18 12 06 1a 1a 1b 04 0c 13 10 11
0x62 ( 98): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x63 ( 99): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x64 (100): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f b8 7f 7f 7f 7f 7f 7f
0x65 (101): 02 18 13 04 0b 03 0e 0f 06 06 12 0d 0f 03 08 05 1d 0f 0f 05 19 0d 18 06 09 14 0f 1e 0d 19 00 07
0x66 (102): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x67 (103): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f d8 7f 7f 7f 7f 7f 7f
0x68 (104): 7f fc 7f 7f 7f 7f 7f b9 7f 7f 9e 7f 7f 7f f8 7f 93 7f 7f bd 7f 7f 7f 7f 7f 7f 7f 7f 7f a4 7f
0x69 (105): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x6a (106): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x6b (107): 1c 07 1c 0b 1b 0a 12 04 09 02 01 0b 10 1a 03 0c 03 14 18 12 04 1d 1b 10 04 0a 03 16 19 09 1e 0d
0x6c (108): 7f 7f 7f c4 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
0x6d (109): 16 0e 0c 04 09 1b 03 06 19 10 17 1d 1e 03 06 03 0e 16 1c 19 13 04 0c 01 01 1a 16 1c 0d 18 0d 00
0x6e (110): 01 1a 05 01 07 04 1a 07 16 07 07 0f 14 1e 11 01 17 09 0e 02 1e 0b 0f 1b 18 1e 11 00 07 1d 05 1c
0x6f (111): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x70 (112): 7f ca 7f 81 7f 7f 7f fa 7f 7f 7f 7f 7f d5 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
0x71 (113): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x72 (114): 09 1d 12 1e 1c 12 0f 14 05 12 01 0c 02 04 1a 0f 0f 0a 1c 0c 0c 11 12 12 08 15 08 01 02 1d 04 09
0x73 (115): 1e 0f 03 15 17 04 00 15 19 11 02 0b 14 13 0f 00 15 09 1a 02 18 1e 02 1a 0e 06 18 18 0d 03 19 19
0x74 (116): 11 1b 06 06 17 10 1b 0d 10 1a 11 0e 1d 0b 1a 04 1a 02 11 09 18 0c 0c 10 1c 0f 15 0f 17 12 16 0a
0x75 (117): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f dd 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
0x76 (118): 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f cd 7f 7f ee 7f 7f 7f 7f
0x77 (119): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x78 (120): 1e 15 0f 17 0c 14 0b 04 1a 14 0d 0a 13 05 10 11 18 0f 06 1a 18 13 07 03 13 1b 07 1e 0f 18 02 12
0x79 (121): 1e 11 09 02 07 0a 1a 1a 1a 1e 0c 1c 1c 02 13 05 0c 16 1c 01 1b 19 0e 07 09 0b 10 09 1c 06 17 0e
0x7a (122): 10 10 0b 0b 01 08 05 03 04 16 07 07 17 0c 00 09 0e 18 0f 0d 0d 0b 1a 1a 09 11 0b 12 16 1e 11 03
0x7b (123): 0a 04 12 10 11 1e 17 12 11 0a 1b 1d 16 18 0a 03 13 1e 10 06 03 09 0a 0a 1a 1c 0f 14 15 0a 12 15
0x7c (124): 00 1b 1c 11 00 1c 0f 19 11 0c 13 14 13 16 0c 00 10 07 17 1d 10 05 0c 17 1a 1c 10 05 1d 0b 08 15
0x7d (125): 1b 19 00 0d 1c 0e 0b 1b 02 1c 08 0e 08 17 02 12 1e 1c 1e 08 02 04 11 09 0d 07 1e 17 0c 01 13 01
0x7e (126): 7f ff 94 7f c7 bf ae 8c ec b7 e4 ce 88 cf f5 c9 f0 d2 e8 b5 e7 7f 90 d4 7f 8e 99 de 87 bb 96 f6
0x7f (127): 7f 7f 7f 7f 7f 7f 9f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f d9 7f 7f 7f 7f
```

50. When accessing virtual address 0x2f9b, what will be the first page accessed (decimal)?
A. 28
B. 78
C. 126
D. 127
E. Error or None of the above
C, access the page directory first.
51. What index of the page directory will be accessed first (hexadecimal)?
A. 0xb
B. 0xce
C. 0x4e
D. 0x1c
E. Error or None of the above
A, top 5 bits of the VA are 01011 = 0xb
52. What will be the second page accessed (decimal)?
A. 11
B. 27
C. 78
D. 113
E. Error or None of the above
C, 78 -- the contents at index 11 (0xb) of the page directory is 0xce. Top bit is valid. Remaining bits are 0x4e (decimal 78)
53. What are the contents of the corresponding PTE that will be read?
A. 0x3d
B. 0xc2
C. 0x11
D. 0x7f
E. Error or None of the above
D, 0x7f -- on page 78 access index 28 to get PTE contents.
54. What is the final physical address for this virtual address
A. 0x7fb
B. 0x4eb
C. 0x3db
D. 0x11b
E. Error or None of the above
E, None of the above. The valid bit is not set.
55. When accessing virtual address 0x0777, what are the contents at the corresponding physical address?
A. 0x03
B. 0x4c
C. 0xf1
D. 0xf8
E. Error or None of the above
A, The physical address is 0xb37 and the value 0x03 is stored at that address.

TLB Hit Rate

The following questions ask you to calculate the miss rate (or hit rate) for the TLB. Assume you have a virtual address that requires 16 bits and there are 512 possible virtual pages per address space. You should ignore all instruction references (i.e., do not consider how they impact the contents of the TLB). Assume the array is page-aligned.

56. Assume you have a 1-entry TLB. Assume the running process sequentially accesses contiguous 4-byte integers in an extremely large array, starting at index 0. What will be the TLB miss rate?

A. 1 / 1
B. 1 / 4
C. 1/ 32
D. 1 / 128
E. None of the above or Not enough information

C, 1/32. Pages are 128 bytes which can hold 32 integers so the first integer on each page will result in a miss.

57. Assume you have a 4-entry TLB with LRU replacement for the same workload. What will be the miss rate in the TLB?

A. 1 / 1
B. 1 / 4
C. 1/ 32
D. 1 / 128
E. None of the above or Not enough information

C, 1/32. The 4 entries in the TLB don't change anything for sequential access (no temporal locality).

58. Assume the running process repeatedly accesses every 4-byte integer in a 128 integer array in a loop (i.e., accesses the elements 0 through 127, then elements 0 through 127 again, over and over). To have a hit rate that approaches 1/1, at least how many entries must be in the TLB? Assume the TLB uses LRU.

A. 1
B. 2
C. 3
D. 4
E. None of the above or Not enough information

D, 4. When all 128 integers can be reached through the TLB the hit rate will approach 1 / 1. The 128 integers require 4 pages.

Page Faults / Swap

A machine has 3 pages of physical memory for a running process. The process accesses pages in the following order: 8, 7, 4, 2, 5, 4, 7, 3, 4, 5.

59. With a FIFO replacement policy, will the second access to page 4 result in a page fault?

A. Yes
B. No
C. Not enough information to determine

B, No. The content of memory will be pages 4, 2, 5 so a page hit will occur (no page fault).

60. With a FIFO replacement policy, what pages will be in memory when the 3rd access to page 4 occurs?

A. Pages 5, 7, and 3
B. Pages 8, 7, and 4
C. Pages 4, 2, and 5
D. Pages 3, 4, and 5
E. Not enough information to determine

A, Pages 5, 7, and 3 will be in memory. One intermediate step: after the second access to page 4, pages 2, 5, 4 will be in memory.