Midterm 1: Virtualization

Time:
Total Points: 70 **Total Questions: 70**

Instructions

This exam is a closed book, closed notes. You will hand the question set and the scantron sheet back to the instructor after the exam is over.

Please fill in your name and student ID on the answer sheet.

When you begin the exam, you will need to agree with the following statements:

1. I understand that this exam must be worked on independently.
2. I will not communicate or collaborate with any others during the exam in any way; for example:
   - *I will not chat, text, or post any questions or information.*
   - *I will not ask for, receive, or provide help to others.*
   - *I will notify the instructor if I am aware of others violating this policy.*
   - *I will not discuss the questions with students of other section.*
3. I will not discuss the questions with students of the other section.

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

- *The OS manages a single uniprocessor (single-core).*
- *All memory is byte-addressable.*
- *The terminology lg means log2.*
- *2^{10}bytes = 1KB.*
- *2^{20}bytes = 1MB.*
- *1 byte = 8 bits*
- *1 hexadecimal digit takes 4 bits*
- *Page table entries require 4 bytes unless otherwise stated.*
- *Data is allocated with optimal alignment, starting at the beginning of a page.*
- *Leading zeros can be removed from numbers (e.g., 0x06 == 0x6).*
- *Hex numbers are represented with a preceding "0x"; numbers without 0x are decimal*

You have seventy-five minutes to complete this exam.

Table of Contents

Section		Question Type	No. of Questions	Total Points
CPU Virtualization		True/False	10	10
		Multiple Choice	10	10
Memory Virtualization		True/False	14	14
		Multiple Choice	11	11
Multi-part	Page Replacement	Multiple Choice	8	8
	CPU Scheduling		4	4
	TLBs		8	8
	Multi-level page tables		5	5

Section 1: In this first section, you will be answering the questions related to CPU Virtualization. The points associated with each question are mentioned at the end of the question.

1. During a system call, the process's registers are saved in the PCB. [1 point]
    a. True
    b. False

2. Return code of fork is always 0 for successful execution. [1 point]
    a. True
    b. False

3. Two processes accessing the same virtual memory address can see different values due to separate address spaces. [1 point]
    a. True
    b. False

4. Response time of processes when using round-robin scheduling is worse than the first-come-first-served (FCFS) scheduling. [1 point]
    a. True
    b. False

5. A program can be associated with more than one process . [1 point]
    a. True
    b. False

6. Exec() will replace the stack and code part of the child process, but not the heap. [1 point]
    a. True
    b. False

7. On a system call trap, the hardware looks up the address of the system call handler in the system call table. [1 point]
    a. True
    b. False

8. A child process created using a fork() system call will start its execution at its main function. [1 point]
    a. True
    b. False

9. In a uniprocessor system, there can be more than one processes in READY and BLOCKED state while at most one process in RUNNING state. [1 point]
    a. True
    b. False

10. When a process is waiting for a read operation to complete from disk, the process will be in the BLOCKED state. [1 point]
    a. True
    b. False

For the following multiple-choice questions, there is only one answer.

11. For FIFO/FCFS scheduling policy, what is the average **turnaround time** for the processes if all of them arrive at the same time? "~" indicates just after the time. [1 point]

Job	Arrival(s)	Run time (s)
A	0	5
B	~0	5
C	~0	5

    a. 10
    b. 20
    c. 5
    d. 20/3

12. For the (Shortest Time-to-Completion First) scheduling policy, what is the average **turnaround time**? "~" indicates just after the time. [1 point]

Job	Arrival(s)	Run time (s)
A	0	30
B	10	5
C	~10	5

a. 25
b. 20
c. 55/3
d. 30

13. Calculate the number of times “hello” is printed in the following program. [1 point]

```
#include <stdio.h>
#include <sys/types.h>

int main()
{
        fork();
        fork();
        fork();
        printf(“hello\n”);

        return 0;
}
```

a. 6
b. 8
c. 4
d. 3

14. Which of these are part of a process but not a program? [1 point]
    a. Code
    b. Global data
    c. main() function
    d. Registers

15. Suppose a round-robin (RR) scheduler with slice times of 5 milliseconds is given three processes to schedule, with arrival times and durations as follows. What is the **average turnaround time**? When a process arrives at the same time a process is preempted, the newly arrived process goes first in the ready queue. [1 point]

Job	Arrival(s)	Duration (milliseconds)
A	0	15
B	0	10
C	5	20

    a. 35
    b. 100/3
    c. 25
    d. 20

16. Which of the following statements is **true**? [1 point]
    a. It is not possible for the user process to directly execute privileged instructions.
    b. While executing a syscall, the process can never be context-switched.
    c. The OS runs in user mode and switches to kernel mode when needed.
    d. Limited Direct Execution allows user programs to directly execute privileged instructions.

17. Under what circumstances a parent and child processes will have the same stack pointer and instruction pointer. [1 point]
    a. When a system call is executed by a parent process.
    b. When a system call is executed by a child process.
    c. When a fork() call is made by a parent process.
    d. When an exec() call is made by a child process.

18. Round robin (RR) and shortest time-to-complete first (STCF) are an example of _____ schedulers, while Shortest Job First (SJF) and First-come-first-serve (FCFS) are an example of _____ schedulers. (The first option in the answers is for the first blank and the second option in the answers is for the second blank.) [1 point]
    a. Non-preemptive & Preemptive
    b. Preemptive & Non-Preemptive
    c. Multi-programming & Uni-programming
    d. Uni-programming & multi-programming

19. A shortest-job-first (SJF) scheduler has the following disadvantages? [1 point]
    a. We may not know how long a process will take until it's already done running.
    b. If a short process arrives later than a long process, it must wait for the long process to finish.
    c. A and B both.
    d. None of the above.

20. Which of the following holds true for multi-level feedback queue (MLFQ) scheduler? [1 point]
    a. One of the goals of a MLFQ scheduler is to support batch and interactive jobs.
    b. With a MLFQ scheduler, a process may preempt another process at the same priority level.
    c. A and B both.
    d. None of the above.

Section 2: You have successfully completed all the questions related to CPU Virtualization. In this second section, you will be answering the questions related to Memory Virtualization. The points associated with each question are mentioned at the end of the question.

21. MMU dynamically translates virtual addresses at every memory reference. [1 point]
    a. True
    b. False

22. The base register contains a virtual address. [1 point]
    a. True
    b. False

23. If the dirty bit is clear for a physical page, then the contents of the page in physical memory match that of the page on the backing store. [1 point]
    a. True
    b. False

24. The page table base register (PTBR) is different for different processes. [1 point]
    a. True
    b. False

25. If we make the page size larger, there will not be any difference in TLB hit rate. [1 point]
    a. True
    b. False

26. Given a large set of data(with, say, a million entries), TLB performance is the same for an array and a linked list. [1 point]
    a. True
    b. False

27. LRU eviction policy will always perform better in TLB as compared to random eviction policy. [1 point]
    a. True
    b. False

28. Reading entries from TLB is slower than reading entries from page tables. [1 point]
    a. True
    b. False

29. LRU (least recently used) page replacement is easier to implement than FIFO replacement. [1 point]
    a. True
    b. False

30. Increasing the size of physical memory always decrease the number of page faults. [1 point]
    a. True
    b. False

31. While using static relocation, a process' address space cannot be moved after it has been placed at a specific location. [1 point]
    a. True
    b. False

32. Whenever a workload accesses n + 1 pages in a loop with one reference per page, where n is the number of TLB entries and using LRU replacement, the miss rate is 50%. [1 point]
    a. True
    b. False

33. Multi-level page tables allow each page table to be allocated non-contiguously. [1 point]
    a. True
    b. False

34. When a page is located on disk and not in memory, its corresponding PTE will have the present bit cleared. [1 point]
    a. True
    b. False

For the following multiple-choice questions, there is only one answer.

35. Where is z located in address space? [1 point]

```
#include <stdio.h>
#include <sys/types.h>

int main()
{
        int *z = malloc(sizeof(int));
        return 0;
}
```

    a. Heap
    b. Stack
    c. Code
    d. No allocation is needed.

36. What are the disadvantages when using base and bound registers? [1 point]
    a. Sharing between two processes is not possible.
    b. Internal & External fragmentation.
    c. Both (a) & (b).
    d. None of the above.

37. Given that we are using segmented addressing, the Base and Bound register for a process P1 are as follows:

Segment	Base	Bounds	R W
0	0x2000	0x6ff	1 0
1	0x0000	0x4ff	1 1
2	0x3000	0xfff	1 1
3	0x0000	0x000	0 0

    Virtual address is 14 bits long and comprises the segment identifier and offset. There are only 4 segments. Convert 0x0256 Virtual Address to Physical Address. Hint: Each hex digit takes 4 bits in an address. [1 point]

    a. 0x0256
    b. 0x3256
    c. 0x2256
    d. 0x4256

38. How big is a page table for the following: for a 16-bit address space (virtual address is 16 bit long) comprising 64-byte pages (page size) and size of page table entry (PTE) is 4 bytes. [1 point]
    a. 1024 (2 ^ 10) bytes.
    b. 2048 (2 ^ 11) bytes.
    c. 4096 (2 ^ 12) bytes.
    d. 8192 (2 ^ 13) bytes.

39. While accessing a multi-level page tables having “N” levels, how many memory references are needed to translate an address? Assume there is no TLB present. [1 point]
    a. 0
    b. N
    c. 2N
    d. 1

40. Given a system using segments and paging with MMU and memory contents below, calculate the correct Physical address for **writing** 0x002070 virtual address: [1 point]

seg $ (4 bits)	page number (8 bits)	page offset (12 bits)

seg	base	bounds	R W
0	0x002000	0xff	1 0
1	0x000000	0x00	0 0
2	0x001000	0x0f	1 1

...
0x01f (Addr – 0x001000)
0x011
0x003
0x02a
0x013
...
0x00c (Addr – 0x002000)
0x007
0x004
0x00b
0x006
...

    a. 0x004070
    b. 0x003070
    c. 0x002070
    d. Error

41. Given a system using segments and paging with MMU and memory contents as shown in previous question (question 40): calculate the correct Physical address for **reading** 0x003080 virtual address: [1 point]
    a. 0x004080
    b. 0x00b080
    c. 0x02a080
    d. Error

42. Suppose a process is run on a system using base and bounds dynamic relocation. The address 0x0000087c (2172 in decimal) is loaded into the base register, and 0x00000800 (2048 in decimal) is loaded into the bounds register. [1 point]

    Which of the following is the physical address, if the virtual address 0x00000652 (1618 in decimal) is requested? [1 point]
    a. 0x00000ece (3790 in decimal)
    b. 0x000016ce (5838 in decimal)
    c. 0x00000a2a (2602 in decimal)
    d. 0x000006ce (1742 in decimal)
    e. None, segmentation violation

43. What is **not true** about the base register? [1 point]
    a. The Base register can be manipulated by user processes.
    b. Only the OS can modify the base register.
    c. The base register contains the starting location
    d. The base register is part of the MMU.

44. To support memory virtualization, _____ and _____ techniques need to allocate the address space contiguously. (The first option in the answers is for the first blank and the second option in the answers is for the second blank.)
    a. Base and Bounds & Segmentation
    b. Segmentation & Paging
    c. Paging & Static Relocation
    d. Static Relocation & Base and Bounds

45. A memory system is a hierarchy of storage devices with different capacities (sizes), costs, and access times. Assume that the lower layers in the hierarchy are further from the processor while higher layers are closer to the processor. As one moves from the higher layers to the lower layers, the capacity (size) _____, while access times _____. [1 points] (The first option in the answers is for the first blank and the second option in the answers is for the second blank.)
    a. Decreases & Increase
    b. Remains same & Decrease
    c. Increases & remains same
    d. Increases & Decrease
    e. Increases & Increase

Multi-Part Questions

You will now have five multi-part questions covering different aspects of virtualization. Most partial questions require you to write in a numerical value within a box; each of these partial questions is graded.

1. **Page Replacement: 8 points**
2. **Scheduling: 4 points**
3. **Base+Bounds Dynamic Relocation: 4 points**
4. **TLBs: 8 points**
5. **Multi-level Page Tables: 6 points**

Page Replacement Policies (8 sub-questions --> 8 points)

Assume the system contains **three pages** (or frames) of physical memory, each of which is originally empty. Assume the following access stream of pages: ABCBDADB

Using the OPT page replacement policy and demand paging, indicate whether each page access (in order) is a hit, miss (including cold), or unknown.

46. A [1 point]
    a. Hit
    b. Miss
    c. Unknown

47. B [1 point]
    a. Hit
    b. Miss
    c. Unknown

48. C [1 point]
    a. Hit
    b. Miss
    c. Unknown

49. B [1 point]
    a. Hit
    b. Miss
    c. Unknown

50. D [1 point]
    a. Hit
    b. Miss
    c. Unknown

51. A [1 point]
    a. Hit
    b. Miss
    c. Unknown

52. D [1 point]
    a. Hit
    b. Miss
    c. Unknown

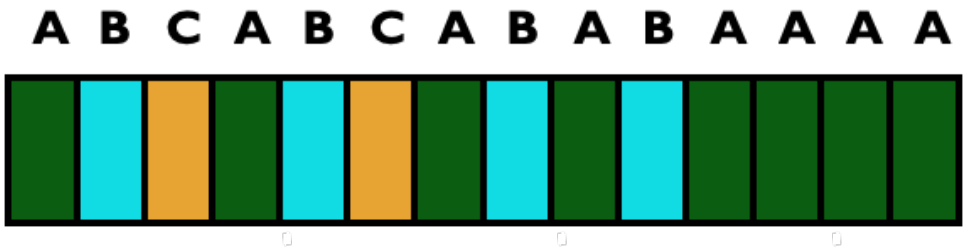
53. B [1 point]
    a. Hit
    b. Miss
    c. Unknown

CPU Job Scheduling [4 sub-parts --> 4 points]

Assume a workload with the following characteristics. You can assume that the jobs arrive in the order mentioned in the table.

Job	Arrival(s)	Run time (s)
A	0	40
B	0	20
C	5	10

54. Which of the schedulers will have the following scheduling timeline:

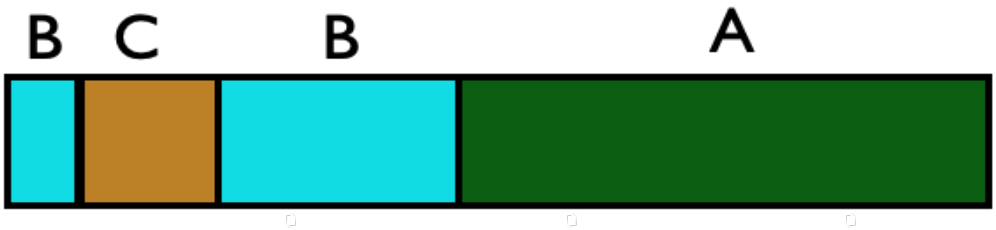


    a. First-in-First out (FIFO)
    b. Shortest Job First (SJF)
    c. Shortest time-completion first / Shortest completion-time first (SCTF/STCF)
    d. Round Robin (RR)

55. Which of the schedulers will have the following scheduling timeline: [1 point]

a. First-in-First out (FIFO)
b. Shortest Job First (SJF)
c. Shortest time-completion first / Shortest completion-time first (SCTF/STCF)
d. Round Robin (RR)

56. Which of the schedulers will have the following scheduling timeline: [1 point]



a. First-in-First out (FIFO)
b. Shortest Job First (SJF)
c. Shortest time-completion first / Shortest completion-time first (SCTF/STCF)
d. Round Robin (RR)

57. Which of the schedulers will have the following scheduling timeline: [1 point]

a. First-in-First out (FIFO)
b. Shortest Job First (SJF)
c. Shortest time-completion first / Shortest completion-time first (SCTF/STCF)
d. Round Robin (RR)

TLB Behavior: Determining TLB contents [8 sub-questions --> 8 points]

Assume a system with the following characteristics:-

- The TLB contains a maximum of 4 entries; the TLB is fully associative with perfect LRU replacement
- Virtual addresses are 16 bits (4 hex digits)
- Physical addresses are 20 bits (5 hex digits)
- Pages are of size 256 bytes
- Instructions are 4 bytes

A user process is executing the following instruction stream that continues for 512 instructions (the address of the instruction is shown in the first column):

0x6000: movl 0x8(%rbp), %edi
0x6004: addl $0x3, %edi
0x6008: movl %edi, 0x8(%rbp)
0x600c: movl 0x8(%rbp), %edi
0x6010: addl $0x3, %edi
0x6014: movl %edi, 0x8(%rbp)
0x6018: movl 0x8(%rbp), %edi
.....

Three instructions continue in this pattern as long as needed for the question. You will note that the first three instructions are just repeated over and over. Assume %rip (the instruction pointer register, or PC or program counter) starts at 0x6000. Assume %rbp (the base pointer register) starts at 0x2000.

Reminders:

- movl 0x8(%rbp), %edi: loads the value at address 0x8 + 0x2000 into the %edi register
- addl: simply adds 3 to the value in the register
- movl %edi, 0x8(%rbp): stores the new value in the register back to memory

Your goal is determine the content of the TLB after 512 instructions execute. The following questions will help you derive the answer.

58. The number of bits in the address used for offset are [1 point]
    a. 4
    b. 6
    c. 8
    d. 10

59. The maximum number of virtual pages are [1 point]
    a. 128
    b. 256
    c. 512
    d. 1024

60. Each page will have how many instructions in a single page. [1 point]
    a. 32
    b. 64
    c. 128
    d. 256

61. How many unique pages are accessed to execute 512 instructions. [1 point]
    a. 4
    b. 8
    c. 16
    d. 32

62. What is the VPN of the first instruction? [1 point]
    a. 0x60
    b. 0x61
    c. 0x62
    d. 0x63

63. What is the VPN of the last instruction? [1 point]
    a. 0x65
    b. 0x66
    c. 0x67
    d. 0x68

64. What is the VPN of the first data access 0x8(%rbp)? [1 point]
    a. 0x10
    b. 0x20
    c. 0x30
    d. 0x40

65. After executing 512 instructions, which VPN will not be present in the TLB? Assume the TLB begins with four INVALID entries. [1 point]
    a. 0x60
    b. 0x65
    c. 0x66
    d. 0x67
    e. 0x20

Multi-Level Page Tables [6 sub-questions --> 6 points]

Consider a system with 32-bit virtual addresses and 4KB pages. PTEs are 4 bytes, and instructions are 4 bytes. Suppose you create a process with these segments:

- Code: 1 MB (1024 KB)
- Global data: 4KB

- Heap: 20 MB (20,480 KB)
- Stack: 64KB

66. Using a linear page the size of the page table will be:
    a. 16-32 KB
    b. 1-3 MB
    c. 4-6 MB
    d. 2-4GB

67. Using paging with segments, the space occupied by all the page tables will be:
    a. 16-32 KB
    b. 1-3 MB
    c. 4-6 MB
    d. 2-4GB

68. Using a standard address space layout, how much space is required for multi-level page tables?
    a. 16 KB
    b. 20 KB
    c. 24 KB
    d. 28 KB
    e. 32 KB

69. Using any possible address space layout (segments can be put at any addresses), what is the *least* amount of space required for a multi-level page table?
    a. 16 KB
    b. 20 KB
    c. 24 KB
    d. 28 KB
    e. 32 KB

70. Consider the instruction, which loads 4 bytes of data at the stack pointer into the EBX register:

    mov %ebx, [%esp]

    Given a multi-level page table with 2 levels and assuming your TLB is empty and a standard process address space, and %ESP = 0x200, how many memory references will be required to execute this instruction?

    a. *2*
    b. *4*
    c. *6*
    d. *8*
    e. *12*