CMPU 334 Quiz1

Fall 2023 Solutions

Name: _____

Instructions:

This is a closed book, closed notes exam. No electronic devices, including calculators, are allowed. You have 75 minutes. There are 10 problems and 10 pages to this exam.

Good Luck!

1. (4 points) Which of the following are valid state transitions for a process? **Circle all that apply.**

   A. Running to Ready

   B. Ready to Blocked

   C. Blocked to Running

   D. Running to Blocked

Valid transitions are A, running to ready and D, running to blocked. B is not a valid transition because a process needs to be running before it can be blocked, because a process gets blocked when waiting for I/O. C is not a valid transition because a process must be in the ready queue before it can run.

2. (7 points) You have the following code shown below. You can assume all calls to `fork` succeed

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main(int argc, char *argv[]) {
    printf("A");
    int rc = fork();
    if (rc == 0) {       // child 1
        printf("B");
        rc = fork();
        if (rc > 0) {    // child 1
            wait(NULL);  // wait for child 2
            printf("C");
        }
        else {           //  child 2
            printf("D");
        }
    }
    else {  // Parent
        wait(NULL);      // wait for child 1
        printf("E");
    }
    return 0;
}
```

(a) (2 points) When the above program is executed, how many processes are created? **__3_____**

There are a total of 3 processes created. The main thread (P1) calls fork, creating child (P2). The child (P2) also calls fork, creating (P3).

(b) (5 points) Which of the following are valid outputs from running the above program? **Circle all that apply.**

A. BADCE

B. ABCDE

C. ABDCE

D. ABDEC

E. ADBCE

3. (3 points) In UNIX, creating a process involves two system calls, `fork` and `exec`. What is the advantage of using two separate system calls instead of one (e.g., `create`)?

4. (6 points) To virtualize the CPU, the operating system relies on Limited Direct Execution for efficiency and control.

(a) (3 points) Explain how the **hardware** is involved to allow the Operating System to regain control when a process is running on the processor.

(b) (3 points) Explain how the **hardware** allows processes to run directly on the processor without letting a process do whatever it wants.

5. (22 points) For the following statements about scheduling, **circle T if the statement is true, and F if the statement is false.**

(a) T / **F** The Shortest Time to Completion First (STCF) scheduling algorithm is optimal with respect to fairness. **False. STCF is optimal wrt turn around time.**

(b) **T** / F The First Come, First Served (FCFS) scheduling algorithm is subject to the convoy effect, where short running processes can queue up behind long running processes. **True**

(c) **T** / F If response time is an important metric to your system, you should use Round-Robin (RR) scheduling. **True. Response time is the elapsed time from when a job enters the system to when it is first run.**

(d) T / **F** In the Multi-Level Feeback Queue (MLFQ) scheduling algorithm, I/O bound processes are moved to lower priority queues because they spend most of their time waiting for I/O anyway. **False. Processes(including I/O bound processes) are kept in their current queue until they use up their entire CPU quanta.**

(e) **T** / F In the Multi-Level Feedback Queue (MLFQ) scheduling algorithm, a process is in exactly one queue at any given time. **True**

(f) T / **F** The Shortest Job First (SJF) scheduling algorithm is optimal with respect to response time. **False. Round Robin gives the best results wrt response time.**

(g) **T** / **F** As the quanta (time-slice) for the Round Robin (RR) scheduling algorithm is decreased, the response-time performance is increased. **This question is worded such that either answer could be correct, depending on how you interpreted the question. When the quanta is decreased, the response time is also decreased. But that could be interpreted as an increase in response time performance, so I'm accepting both answers.**

(h) **T** / F The Shortest Time to Completion First (STCF) scheduling algorithm is only possible in an operating system that supports preemption. **True. When a new job enters into the system, it may be shorter than the remaining time on the current job. In this case, the OS needs to preempt the current process to run the new process.**

(i) **T** / F The Multi-Level Feedback Queue (MLFQ) scheduling algorithm runs processes at the same priority level in a Round Robin fashion. **True**

(j) T / **F** In the Multi-Level Feedback Queue (MLFQ) scheduling algorithm, processes start at the lowest priority until it is determined whether the process is CPU or I/O bound. **False. Processes start at the highest level queue, and move down once they use up all the time in their quanta for that level.**

(k) T / **F** In the Multi-Level Feedback Queue (MLFQ) scheduling algorithm, Priority Boot makes sure that a process can not game the system by giving up the CPU right before its time-slice ends. **False. The priority boost prevents starvation, by periodically raising processes in the lower queues to the highest queue**.

6. (10 points) For the following statements about the Translation Lookaside Buffer (TLB), **circle T if the statement is true, and F if the statement is false.**

(a) **T** / F TLB misses are handled by the hardware in the Intel x86 architecture. **True**
.

(b) **T** / F Using an address space identifier (ASID) in the TLB prevents having to flush the TLB on a context switch. **True**

(c) T / **F** The TLB reduces the size of the page table, at the cost of more memory accesses to perform address translation. **False. The TLB is a cache of popular page table entries. It has nothing to do with the size or organization of the page table.**

(d) T / **F** Protection bits are not stored in the TLB because they are stored in the Page Table Entry (PTE) for a page. **False. This information must also be kept in the TLB because in the case of a TLB hit, the page table is never accessed.**

(e) **T** / F Increasing the page size for a system would lead to fewer TLB misses. **True. A larger page size means each TLB entry has more memory addresses that refer to that entry.**

7. (10 points) Which of the following are policies, and which are mechanisms? **For each answer below, circle either policy or mechanism**.

(a)	Performing a context switch	Policy / **Mechanism**
(b)	Using a Round-Robin scheduler to select a process to run	**Policy** / Mechanism
(c)	Having a system call trap into the OS	Policy / **Mechanism**
(d)	Using the Least Recently Used (LRU) algorithm to find a page to swap to disk.	**Policy** / Mechanism
(e)	Walking a page table to find a Page Frame	Policy / **Mechanism**

8. (8 points) Which of the following design decisions can lead to more internal fragmentation or more external fragmentation? **For each answer below, circle either Internal or External.**

(a)	Increasing the page size for a system using page tables	**Internal** / External
(b)	Using segmentation instead of page tables for virtual memory	Internal / **External**
(c)	Using a base register and a bounds register to map a virtual address space to a physical address space	**Internal** / External
(d)	Using a hybrid system with a page table for each segment along with a bounds register for each segment marking the number of pages in the segment.	Internal / **External**

9. (12 points) For the following statements, **circle T if the statement is true, and F if the statement is false.**

(a) T / **F** A multi-level page table is typically much larger than a linear page table. **False. The purpose of a multi-level page table is to reduce the size of a linear page table. If an entire page worth of page table entries are not valid (in use), that can be recorded as a bit in the page directory, and that page of the page table entry does not need to exist.**

(b) T / **F** A multi-level page table speeds up page table entry (PTE) lookups in the event of a TLB miss. **False. A memory lookup has to walk the page table. Each level in the page table requires another memory access.**

(c) **T** / F A trap causes a switch from user mode to kernel mode. **True**

(d) **T** / F The clock algorithm tries to approximate an LRU algorithm but with lower overhead. **True**

(e) **T** / F The VAX/VMS virtual memory system is an example of a hybrid system. **True**

(f) **T** / F Each process has its own page table. **True. This is the mechanism that isolates a process's memory address from other processes.**

10. (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 18 bits wide.
- Physical addresses are 18 bits wide.
- The page size is 1024 bytes.
- The TLB is 8-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	62	C6	1
	44	48	1
	36	58	0
	0C	3D	1
	3B	8D	0
	24	9E	1
	35	80	0
	08	9A	1
1	15	93	0
	13	A3	1
	0C	61	0
	24	75	1
	4C	CD	0
	2E	E1	0
	36	9C	1
	08	81	0

Page Table (first 32 pages only)					
VPN	**PFN**	**Valid**	**VPN**	**PFN**	**Valid**
00	8B	1	10	84	1
01	D8	0	11	70	1
02	E4	1	12	09	0
03	AC	0	13	23	0
04	A4	1	14	61	0
05	E4	0	15	9E	1
06	2E	1	16	8F	0
07	03	0	17	A1	1
08	58	1	18	A0	1
09	1F	1	19	25	1
0A	8B	1	1A	75	0
0B	C9	1	1B	CA	1
0C	29	0	1C	11	0
0D	5E	1	1D	09	0
0E	D8	0	1E	33	0
0F	BF	1	1F	8C	1

(a) (3 points) How many total **virtual** pages (VPNs) are there in this system? ____**256**_____

(b) (3 points) How many total **physical** frames (PFNs) are there in this system? __**256**_____

(c) (6 points) Translate the given virtual address below into its physical address. Indicate the virtual page number (VPN) of the virtual address, TLB tag and index bits, and whether the TLB lookup is a hit or a miss. If the memory access is valid, give the physical frame number (PFN), and the physical address of the translation. If the virtual address is not valid, leave the PFN and Physical address fields blank.

Virtual address : **0x092D6**

Parameter	**Value**
VPN	**0x24**
TLB Index	**0x0**
TLB Tag	**0x12**
TLB Hit? (Y/N)	**N**
PFN	**0x**
Physical Address	**0x**

(d) (6 points) Translate the given virtual address below into its physical address. Indicate the virtual page number (VPN) of the virtual address, TLB tag and index bits, and whether the TLB lookup is a hit or a miss. If the memory access is valid, give the physical frame number (PFN), and the physical address of the translation. If the virtual address is not valid, leave the PFN and Physical address fields blank.

Virtual address : **0x04453**

Parameter	**Value**
VPN	**0x11**
TLB Index	**0x1**
TLB Tag	**0x08**
TLB Hit? (Y/N)	**N**
PFN	**0x70**
Physical Address	**0x1C053**