CS 347: Operating Systems, Autumn 2018

Mid-semester Examination,, 10th September 2018

Name: ——————————————— **Roll No.:** ———————————

A **Maximum marks: 30**, **Duration: 2 hours**

- This exam is closed books and notes, closed phones and laptops, and closed friends.

- As part of answers, state assumptions explicitly, if any.

---

1. **True/False**. No justification, no marks. Reverse statements are not justifications.

   (a) If an operating system uses the copy-on-write optimization as part of the `fork` call, it cannot use *lazy allocation scheme* to manage memory allocations of processes.

   (b) A multi-CPU machine has as many CR3 registers as the number of CPUs.

   (c) A page table entry has all the information required to handle a page fault on accessing a swapped out page. Note: The MMU does not provide reasons for a page fault.

   (d) In `xv6`, `PHYSTOP`, the maximum mapped physical memory, is set to 224 MB and `NPROC`, the maximum number of processes, is set to 64. Increasing `NPROC` to accommodate more processes will require increasing `PHYSTOP` as well.

   (e) A process creates a child process using `fork`, and immediately exits. If the child cannot issue `exec` before parent exits, the child process will never be able to load a new program.

   (f) A memory leak—allocated memory which cannot be discarded because no reference to it exists in the user-level process, is not a problem for memory management after a process exits.

   (g) As part of the `malloc` call implementation, TLB entries of the CPU will **have** to be updated for newly added mappings to the page table (assume no lazy allocation).

   (h) The operating system scheduler function is invoked on the timer interrupt and never as part of a system call.

   (i) A partial TLB flush (removal of a subset of the TLB entries) during a process context switch will always hamper correct execution of processes. Assume no `ASID` support.

   (j) If a SIGINT signal is delivered to a process (via the PCB) by the kernel, the process will not be able to generate any interrupt (implicit or explicit).

2. Consider a parent process that has forked two child processes as follows.

```
int x = 10;
if(fork() == 0) {          // Child process 1
    x++;
    printf("A = %d \n", x);
    exit();
}
x--;
if(fork() == 0) {          // Child process 2
    x += 10;
    printf("B = %d \n", x);
    exit();
}
x += 5;
printf("C = %d \n", x);
exit();
```

   (a) What are the values printed for the variables A, B, and C, if the scheduler always schedules the parent process before the child process after the fork call? Explain. (1.5 marks)

   (b) What are the values printed for the variables A, B, and C, if the scheduler schedules the parent process after the child process exits? Explain. (1.5 marks)

3. Assume 20 bytes are available to store user level pages and the page size is 4 bytes. For the following user-level virtual address accesses, estimate the number of page faults for the LRU, FIFO and Optimal replacement policies.
Virtual address access sequence is as follows: 10, 30, 5, 11, 35, 3, 20, 11, 29, 34, 10, 23, 4, 0, 8. (4 marks)

4. Explain how the following works using a diagram,
Process P1 issues a system call, and from within the system call the kernel updates some kernel *state*. The same updated *state*, is visible when the kernel accesses it via a system call from another process P2 or from an interrupt handler. (3 marks)

5. Consider a system where TLB look up time is 25 ns and memory access time is 200 ns, respectively. Assuming a virtual address space of 2 KB, page size of 32 bytes, and a PTE size of 2 bytes, what is the minimum TLB hit ratio that results in an average v2p translation latency of 185 ns? (3 marks)

6. Operating systems disable interrupts while executing an interrupt handler. Interrupts are enabled as the last action of the handler. Explain with an example why this is required? (2 marks)

7. Explain with a diagram what is the maximum number of pages required to store the page table for mapping 1024 memory accesses of a 32 bit virtual address space, with 4 byte PTE entries, and 4 KB pages.
Assume that a process has access to its entire address space, unlike with xv6 which reserves the upper portions of an address space for the kernel. (2 marks)

8. Describe with a flow-chart/diagram the sequence of operations to handle a page fault. Note that page faults can occur due to different reasons. (3 marks)