CSE Ph.D. Qualifying Exam, Spring 2020
Systems & Architecture

PRINT NAME: _____

PRINT PSU EMAIL: _____

PSU ID: SIGNATURE: _____

INSTRUCTIONS: **Do 6 of 8 problems.** ***Indicate on the grading table below which problem is NOT to be graded by crossing out the corresponding column in the grade box below*****, otherwise an arbitrary subset of 6 questions will be chosen for grading.** No aids allowed. Neatly and coherently **justify** your answers in the space allotted. Use the back of a page or facing page if necessary and, if you do, clearly mark "continue on back" or "continue on facing page."

QUESTION:	1	2	3	4	5	6	7	8	TOTAL
SCORE:									
MAXIMUM:	15	15	15	15	15	15	15	15	90

1. (15 pts) Consider the following questions about UNIX filesystems.

(a) (3pts) What is the absolute pathname of a file "/foo/bar.exe" in a filesystem partition mounted at the directory "/boo/bird"?

(b) (3pts) Suppose a process makes a request to open the file "/usr/bin/ls". List in order the inodes that must be accessed to open this file.

(c) (3pts) Assume 1000-byte disk blocks. Suppose that each directory entry consumes 100 bytes. Suppose in "bar.exe" is the 35th directory entry in the directory "foo". How many data blocks of the "foo" inode must be accessed to find the file "bar.exe"?

(d) (3pts) Assume 1000-byte disk blocks. Suppose that we want to read a byte at the offset 18,200 in the file "bar.exe". How many data blocks (discount meta-data blocks) for that file must be accessed to locate and read that byte?

(e) (3pts) Suppose a process invokes `open` on the file path "foo/bar.exe" and receives a return value of 7. How does that value enable the process to use the corresponding file inode?

2. (15pts) Answer the questions below based on the page mappings shown below. Note that this is a logical representation of the contents of multiple page tables. Assume 4K pages. Page and frame numbers are in decimal format.

Process #	Page #	Frame #	Ref Bit	Dirty Bit	Perms
2	8	11	1	0	r
1	5	7	1	1	rw
3	9	6	0	1	rw
1	11	18	0	1	rw
3	8	10	1	0	rx
2	2	32	1	1	r
2	9	19	0	1	rw
1	20	12	1	0	rx

(a) (3pts) What physical memory address is accessed when process 1 accesses the virtual address 0xb321?

(b) (3pts) Suppose one of the processes references a page that causes a page replacement to be initiated. Using the *clock algorithm*, starting at the top of the list (clock index is at top of list), which page table entry (page and frame) would be replaced?

(c) (3pts) Write the resulting page table entry for the new page assuming that process 3 performed a read operation at address 0x1511f to cause the replacement (minimal page permissions for the operation)?

(d) (3pts) Suppose that process 3 tries to write at address 0x2220. What will be the result of this operation?

(e) (3pts) Suppose that a process consumes 4MB of virtual memory and has a maximum

working set size of 2MB. Further, suppose that the process runs on a system with a *two-level page table* using 8 bits for the page directory and 12 bits for addressing the page table entries. What is the minimal amount of page table memory needed by the process, assuming 10 bytes per entry?

3. (15pts) Answer the following questions about the following pseudocode. The pseudocode has been written with the intention of realizing a "barrier" - each thread has a special instruction within its code beyond which the thread's execution may proceed only once all threads have executed their respective special instructions. Before any thread executes the code assume: (1) mutex is a semaphore initalized to 1; (2) barrier is a semaphore initialized to 0; (3) count is initially set to 0; and (4) n is the number of threads.

Prog X

```
1    initialize
2    mutex.wait()
3    count = count + 1
4    mutex.signal()
5    if count == n: barrier.signal()
6    barrier.wait()
7    critical point
```

(a) (7pts) Explain why this pseudocode fails to correctly realize the barrier pattern.

(b) (8pts) Specify a fix to realize the correct barrier pattern.

4. (15pts) Consider the following *preemptive multilevel queue scheduler* and process information. A quantum for a process only restarts on an I/O event.

   - Queue Q_0 (Priority 2, Quantum 4): Schedule Processes in Preemptive Shortest Job First and Demote to Q_1 if use quantum
   - Queue Q_1 (Priority 1, Quantum 6): Schedule Processes in Round Robin and Promote to Q_0 if not use quantum

   Q_1 has higher priority than Q_0.

Process ID	Arrival Time	CPU Burst Time	Priority (Queue)
P1	0	4	1
P2	2	3	2
P3	3	5	2
P4	6	4	1
P2	10	2	?

(a) (3pts) What is the *demotion policy* for this multi-level feedback queue scheduler?

(b) (3pts) If P1 is running at time 2, what action is taken by the scheduler when P2 becomes ready?

(c) (3pts) When is P3 first scheduled and when is it first de-scheduled?

(d) (3pts) What queue is P3 in at time 10?

(e) (3pts) Which process is running at time 10 (or starts at time 10)?

5) Datapath Design (15 pts)

a. Label all data dependencies and (false/anti)-dependencies in the below code fragment with an arrow and their data dependency type. (5 points)

```
F00: lw     $2, 0($4)
lw          $3, 0($5)
addu        $2, $2, $3
sw          $2, 0($4)
lw          $4, 4($4)
addi        $5, $5, 4
bne    $4, $0, F00
```

b. What are types of dependencies that limit degree of Instruction Level Parallelism? Suggest at least one technique to overcome these dependencies. (4 points)

c. (6 pts) Consider the following piece of code:

```
addi    $1, $0, 32
 add    $2, $0, $0
 ENTOUTER:
 addi   $3, $0, 4
 ENTINNER:
 mul    $4, $1, $5
 add    $6, $4, $3
 lb     $7, 0($6)
 addi   $3, $3, -1
 add    $2, $2, $7
 beq    $7, $0, EXTINNER    ; branch 1
 bne    $3, $0, ENTINNER    ; branch 2
 EXTINNER:
 addi   $1, $1, -1
 bne    $1, $0, ENTOUTER    ; branch 3
 EXTOUTER:
```

For simplicity, assume that the data loaded by the lb instruction is always either 0 or 1 and that the first 8 loaded values are 10100110

(i) Using T for taken and N for not taken, what are the directions of the first 10 branches encountered?

(ii) Assuming a static prediction scheme that predicts always-taken for backwards branches and always-not-taken for forwards branches, what is the accuracy of said scheme on the first 10 branches encountered?

(iii) What is the prediction accuracy of a dynamic 2-bit (N, n, t, T) saturating counter branch prediction scheme for the first 10 branches if the initial state is (t)?

(iv) Assume a local counter for each static branch (with no aliasing) using the above counter scheme. What is the new accuracy?

6. Cache/Memory System (15 point)

a. Suppose we have a processor with a base CPI of 1.0, assuming all references hit in the primary cache, and a clock rate of 5 GHz. Assume a main memory access time of 100ns, including all the miss handling. Suppose the miss rate per instruction at the primary cache is 2%. How much faster will the processor be if we add a secondary cache that has a 5ns access time for either a hit or a miss and is large enough to reduce the miss rate to main memory to 0.5%? (4 points)

b. Assume a cache of 4K blocks, a four-word block size, and a 32-bit address, find the total number of sets and the total number of tag bits for caches that are direct-mapped, two-way and four-way associative and fully-associative. (4 points)

c. Show an example of cache accesses that will increase cache misses when moving from direct mapped to fully associative caches (when keeping cache size constant). (3 points)

d. We have a virtual memory system that has a physically-indexed, physically tagged cache. A memory reference can encounter three different types of misses: a TLB miss, a page fault and a cache miss. Consider all possible combination of these three events with one or more occurring. For each possibility, state whether this event can actually occur and under what circumstances. (4 points)

7. Performance Analysis (10 points)

a. An image processing task takes 30% of the overall processing time of an application. This task is now accelerated in a GPU. (5 points)

   i. What is the maximum speedup that can be expected for the entire application through this acceleration?

   ii. What are constraints that will limit approaching this maximum speedup (List at least 3]?

b. Most processors are designed to have adjustable voltage, so a 15% reduction in voltage may result in a 15% reduction in frequency. What would be the impact of dynamic energy and on dynamic power? (5 points)

8) Parallelism (10 points)

(a) Consider the loop below

```
for (i=0; i <100; i=i+1) {
   A[i[ = A[i] + B[i]; /* S1 */
   B[i+1] = C[i] + D [i]; /* S2 */
}
```

What are dependences between S1 and S2? Is this loop parallel? If not, restructure code to make it parallel. (3 points)

(b) Suppose you want to achieve a speedup of 80 with 100 processors? What fraction of the original computation can be sequential? (3 points)

(c) Suppose we have an application running on a 32-processor multiprocessor, which has a 200ns time to handle reference to a remote memory. For this application, assume that all the references except those involving communication hit in the local memory hierarchy, which is slightly optimistic. Processors are stalled on a remote request and the processor clock rate is 3.3GHz. If the base CPI (assuming all reference hit in the cache) is 0.5, how much faster is the multiprocessor if there is no communication versus if 0.2% of the instruction involve a remote communication reference? (4 points)