CPSC/ECE 3220 – Summer 2017 – Exam 2 Name: _____

**Part 1: Word Bank**

Write one of the words or terms from the following list into the blank appearing to the left of the appropriate definition. Note that there are more words and terms than definitions. (1.5 pts. each)

| | | | |
|---|---|---|---|
| affinity scheduling | fine-grained locking | mutually recursive locking | real-time scheduling |
| base and bound | gang scheduling | nested waiting | scheduler activations |
| compute-bound task | I/O-bound task | oblivious scheduling | TLB hit |
| core map | lock-free data structure | preemption | TLB miss |
| dynamic priority scheduling | lock ordering | priority donation | TLB shootdown |
| false sharing | long-term scheduler | processor scheduling policy | zero-on-reference |

1. _____ A task that primarily does I/O, and does little processing.

2. _____ A task that primarily uses the processor and does little I/O.

3. _____ A request to another processor to remove a newly invalid TLB entry.

4. _____ A widely used approach to prevent deadlock, where locks are acquired in a pre-determined order.

5. _____ An early system for memory protection where each process is limited to a specific range of physical memory.

6. _____ When there are more runnable threads than processors, the policy that determines which threads to run first.

7. _____ A scheduling policy where the OS assigns threads to processors without knowledge of the intent of the parallel application.

8. _____ A scheduling policy for multiprocessors that performs all the runnable tasks for a particular process at the same time.

9. _____ A scheduling policy where tasks are preferentially scheduled onto the same processor that they had previously been assigned to improve cache reuse.

10. _____ When a thread waits for a lock held by a lower priority thread, the lock holder is temporarily increased to the waiter's priority until the lock is released.

11. _____ A multiprocessor scheduling policy where each application is informed of how many processors it has been assigned and whenever the assignment changes.

12. _____ A data structure used by the memory management system to keep track of the state of physical page frames, such as which processes reference the page frame.

**Part 2: Multi-Object Locking and Non-Blocking Synchronization**

Compare and Swap / Mellor-Crummey Scott lock / Read-Copy-Update. Circle **only one** of CAS, MCS, or RCS. (1.5 pts. each)

13. CAS / MCS / RCU Has a "grace period".
14. CAS / MCS / RCU Is an atomic read-modify-write instruction.
15. CAS / MCS / RCU Is an efficient form of readers/writer locking.
16. CAS / MCS / RCU An efficient spinlock implementation where each waiting thread spins on a separate memory location.

17. Suppose you have a table of 64 counters shared by hundreds of threads on a multiprocessor with 8 processors. Each counter is 4 bytes in length, and a cache line is 64 bytes in length. Updates (i.e., write accesses) to the counters account for approximately two-thirds of the accesses, and read-only accesses account for the rest. There is no discernable pattern in which a thread accesses a particular counter, so consider the accesses as random. Identify any correctness and/or performance concerns you have for the following approaches during peak times when dozens of threads may contend for access to any given lock: (8 pts.)

    (a) a single mutual exclusion lock for the whole table
    (b) a separate mutual exclusion lock for each counter
    (c) an RCU lock for the whole table
    (d) CAS-based optimistic concurrency control used for counter updates, with no synchronization for reads

**Part 3: Deadlock**

18. Give the four necessary conditions for deadlock. (6 pts.)

True/False. Circle **only one** of T or F. (1 pt. each)

Use this table for questions 19-22 for deadlock avoidance questions based on the Banker's algorithm. Treat each question independently, starting each time from the values shown in the table.

| process | max need | allocated | remaining need |
|---|---|---|---|
| A | 10 | 4 | 6 |
| B | 9 | 2 | 7 |
| C | 8 | 3 | 5 |

Unused units = 6

19. T / F Can safely grant request of process A for 1 units.

20. T / F Can safely grant request of process B for 1 unit.

21. T / F Can safely grant request of process B for 2 units.

22. T / F Can safely grant request of process C for 2 units.

23. T / F An unsafe state will always lead to deadlock.
24. T / F If you prevent deadlock by making sure that one of the four conditions does not hold, you will also prevent starvation.

**Part 4: Scheduling**

Short-Term/Medium-Term/Long-Term Scheduling. Circle **one or more** of S, M, L, as applies. (1.5 pts. each)

25. S / M / L Is called a dispatcher.
26. S / M / L Typically runs after every interrupt.
27. S / M / L Attempts to dynamically maintain a balanced job mix.

FIFO/RR/MFQ/SJF-preemptive. Circle **one or more** of F, R, M, S, as applies. (2 pts. each)

28. F / R / M / S Is preemptive.
29. F / R / M / S Allows starvation.
30. F / R / M / S Uses time quantums.
31. F / R / M / S Requires future knowledge.
32. F / R / M / S Has minimum average response time.

True/False. Circle **only one** of T or F. (1 pt. each)

33. T / F A program is the unit of scheduling in a computer system.
34. T / F A CPU timer allows round-robin scheduling to be implemented.
35. T / F SJF-preemptive is used by most time-sharing systems.

36. Given the following list of tasks, arrival times, and service times, calculate the completion (i.e., departure) time and response (i.e., turnaround) time of each task for FIFO scheduling and again for SJF-preemptive scheduling. (6 pts.)

| | | | ---------- FIFO ---------- | | ------ SJF-preemptive ------ | |
|---|---|---|---|---|---|---|
| Task | Arrival Time | Service Time | Completion Time | Response Time | Completion Time | Response Time |
| A | 0 | 10 | _____ | _____ | _____ | _____ |
| B | 2 | 4 | _____ | _____ | _____ | _____ |
| C | 4 | 1 | _____ | _____ | _____ | _____ |

Aging/Boosting. Circle **only one** or A or B. (1 pt. each)

37. A / B Priority increases while waiting.
38. A / B Priority decreases while running.
39. A / B Priority increase value specified in signal operation for I/O completion.

40. Devise a workload when FIFO is pessimal for average response time. (2 pts.)

41. Explain the priority inversion problem. (4 pts.)

**Part 5: Address Translation**

Base and Bound relocation registers/Segmentation/Paged memory. Circle **one or more** of BB, S, or P, as applies. (1.5 pts. each)

42. BB / S / P Uses variable-length allocation.
43. BB / S / P Can have internal fragmentation.
44. BB / S / P Can be sped up by the use of a TLB.
45. BB / S / P Performs a bounds/limit check on the address.
46. BB / S / P Process control block holds the base address of the mapping table.

The MULTICS system was word-addressable, and a virtual address was structured in this manner:

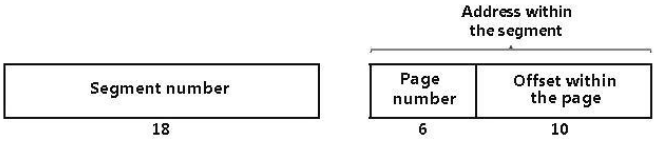


**Figure 3. A 34-bit MULTICS virtual address.**

Answer questions 47-48 using powers of 2. Use words as the addressable units, and do not convert to bytes.

47. Consider a data structure of 512 K words. How many segments do you need to store this structure? (1 pt.)

48. How many pages do you need to store the data structure in question 47? (1 pt.)

Consider a simple architecture with paged segments in which three-decimal-digit virtual addresses are used. The segment id is the first decimal digit, the virtual page number is the second decimal digit, and the offset is the last decimal digit. The physical address are two decimal digits: the page frame is the first decimal digit, and the offset is the second decimal digit. E.g., the virtual address 013 would be translated to physical address 23 given the following segment and page table entries. Give the physical addresses for the following virtual addresses. (2 pts. each)

| **Segment Table** | **Page Table A** | **Page Table B** |
|---|---|---|
| 0 = points to page table A, segment length is 3 pages | 0 = page frame 4 | 0 = page frame 7 |
| 1 = points to page table B, segment length is 2 pages | 1 = page frame 2 | 1 = page frame 5 |
| X (other values are invalid) | 2 = page frame 0 | X (other values are invalid) |
| | X (other values are invalid) | |

49. Virtual address 000 is translated to physical address _____.

50. Virtual address 111 is translated to physical address _____.

51. What should happen when the processor tries to translate virtual address 123 and why?

52. Explain what the term “superpage” means and how the concept helps improve TLB performance. (2 pts.)

53. Explain what is meant by a “tagged TLB” and how the concept helps improve TLB performance. (2 pts.)

54. Consider the following two approaches to zero matrix “a”. Assume that matrix a is stored in row-major order, that a double-precision value is 8 bytes, that the page size is 4 KB, that the i and j index variables are register-allocated rather than memory allocated, and that you have a traditional TLB with 8 entries. For each approach, compute the number of TLB misses, assuming that for each approach the TLB starts off as empty. (4 pts.)

```
double a[1024][1024];

for( i = 0; i < 1024; i++ ){
        for( j = 0; j < 1024; j++ ){
                a[i][j] = 0.0;
        }
}
```

```
double a[1024][1024];

for( j = 0; j < 1024; j++ ){
        for( i = 0; i < 1024; i++ ){
                a[i][j] = 0.0;
        }
}
```

Extra Credit. Identify the two distinct ways to recover from deadlock mentioned in the textbook. (up to 3 pts.)