OMSCS CS 8803 Intro to OS Spring 2016 Final Exam Solution

**Q1. [10 points]**

A round robin scheduler with timeslice value of 2ms schedules an I/O bound task and two CPU bound tasks. The I/O bound task repeats 5 times a loop where it computes for 1ms and then issues an I/O request, where each I/O operation takes 1ms to complete. Each of the CPU bound tasks take 10ms. The context switching overhead is 0.1ms. Assume that the initial order in which these tasks are scheduled starts with the I/O task, followed by the CPU tasks.

For this system, complete the following:

-       [2pt] Total execution time in ms:

-       [3pt] average throughput in tasks in ms:

-       [2pt] average wait time in ms:

-       [3pt] average CPU utilization, rounded up to the nearest %:

1 I/O w/ 5 requests, 2 CPU with 10ms, t=2ms

- x - - x - - x - x - - x - - x – x - - x - - x – x - - x - - x – x - - x - -

t = (5 + 10 + 10 + 14\*0.1) = 26.4

T = 3/t = 0.11 tasks/ms

t\_wait =  (0 + 1.1 + 3.2) / 3 = 1.43

CPU\_util = 25/26.4  = 0.946 = 95%

**Q2: [5 points]**

Enter one of the words ‘higher’ or ‘lower’ to complete the following sentences:

1.     In the Linux O(1) scheduler, a \_\_\_\_\_\_\_\_\_\_\_ priority thread is assigned larger time quantum.

2.     In a traditional Multi-level Feedback Queue Scheduler, a \_\_\_\_\_\_\_\_\_ priority thread is assigned to a queue with larger time quantum

Answer: higher (2.5pt), lower (2.5pt)

**Q3. [10 points]**

In the paper Chip Multithreading Needs a New OS Scheduler, the authors explore ways for the OS scheduler to use hardware counters to extract information about the types of tasks in the system.  Based on the synthetic workload that is analyzed, which of the following are true when scheduling 2 tasks on the hardware threads of a single core:

a)     (A) Scheduling on the same hyperthreaded core 2 tasks with mixed CPI values leads to high IPC

b)    (B) Scheduling on the same hyperthreaded core 2 tasks with low CPI values leads to high IPC

c)     (C) Scheduling on the same hyperthreaded core 2 tasks with high CPI values leads to high IPC

Mark all that apply.

Answer: correct: A and B (4 each), incorrect C (not marked == 2pt)

-       one thread in A is CPU bound, so will keep running and issuing instructions, so the IPC will be high. That was the whole point of the paper, to schedule 2 tasks with a mix of CPI values so as to maximize the IPC.

-       Both threads in B are CPU bound (low CPI), one of them will always get to issue instructions, so the overall IPC will be high.

**Q4. [10 points]**

From the list below, select only the mechanisms used by OSs that allow an application to allocate and reference more memory than the size of the physical memory available on the machine:

1. virtual address translation

2. demand paging

3. least recently used cache replacement

4. buddy allocator

answer: address translation, demand paging

3pt for each correct, 2 points for not marking the incorrect ones.

**Q5. [5 points]**

Shared-memory-based communications try to avoid which of the following overheads, when compared to message-based IPC: (mark all that apply)

|  |  |
| --- | --- |
| A. (A) context switching among sender and receiver if running on a single core |  |
| B. (B) mapping of shared memory into an address space |  |
| C. (C) system calls |  |
| D. (D) data copying | |

Answer: C and D; 2 each correct, 0.5 each for not marking an incorrect.

**Q6. [5 points]**

You need to implement synchronization of readers and writers contending for a shared file. The file may be accessed by up to 10 readers at a time, or by a single writer. You are told that the only type of synchronization construct you may use are semaphores.

Which design seems best suited for this problem:

a)     (A) Use one semaphore initialized to 10

b)    (B) Use one semaphore initialized to 1

c)     (C) Use one semaphore initialized to 10, and one semaphore initialized to 1

d)    (D) Use two semaphores, both of them initialized to 1

answer: (B), the other ones won’t work, shared resource can only be protected with a single mutex/semaphore if you want to be able to make any guarantees that a writer won’t access it at the same time as other readers.

**Q7. [10 points]**

The Anderson Spinlock paper points out that there are three key metrics to analyze the performance of spinlock implementations: latency, delay and contention.

Use the above terms complete the following sentences:

1. Test\_and\_test\_and\_set (or spin on read) has better {latency} than the queuing lock.

2. Test\_and\_test\_and\_set has worst {contention}.

3. Under light loads, static delay algorithms have worse {delay} than dynamic delay algorithms.

4. The queuing lock has better {delay|contention} than the delay-based locks.

2.5 each.

**Q8. [10 points]**

Consider a file system whose inode structure include 10 direct block pointers, 1 single-level indirect pointer and 2 double indirect pointers. Each block pointer is 4 bytes. Each block is 2 kilobytes.

Answer the following:

a)     [2.5 pt] how many blocks can be addressed by the direct pointers: 10 \_\_\_ blocks

b)    [2.5 pt] how many blocks can be addressed by the single-level pointer: 2kB/4 =2^11/2^2 =2^9 = 512 blocks (will accept 500)  \_\_\_ blocks

c)     [2.5 pts] how many blocks can be addressed by each of the double-level pointers: 512 \* 512 = 2^18 = 256 kiloblocks = 262144 \_\_\_ blocks (will accept 250000)

d)    [2.5 pts] what is the maximum size of a file in this file system, rounded to the nearest GB: 2\*2^18 \* 2^11 = 2^30 = 1  \_\_\_ GB

Note: for your answers in b) and c), you may leave the answer in a form 2^n, where n is an integer.

**Q9. [5 points]**

Consider how the protection rings and modes present in x86 architectures are used in virtualized systems.

Which of the following requirements is/are addressed by running user processes, guest OSs and hypervisors in different protection rings or modes.

Check all that apply.

a.      (A) Protect hypervisors from guest OSs

b.     (B) Protect guest OSs from their applications

c.      (C) Protect guest OSs from one another

d.     (D) Protect guest applications from one another

Answer: A and B correct +2 each; none of C and D: +0.5 each

**Q10. [5 points]**

For the following XDR definition, what is the length, in bytes, of an array of 10 32bit integers, when transmitted during an RPC call.

occupied by arguments of this type when transmitted during an RPC call.

a) typedef int data[10];

answer: 40

b) typedef int data<>;

answer: 44 (accepted 48)

**Q11. [5 points]**

A common solution in distributed data stores (or distributed file system) is to organize the metadata in two types of data structures: a global map (or global index) – to identify a specific node in the system, and a data map – to identify the location of the contents (e.g., chunks or blocks) of the replicated or partitioned data item (or file). What is the common way of implementing these data structures:

Global Map: replicated (B)

Data Map: partitioned (A)

2.5pts each

**Q12. [5 points]  - 5x1pt questions.**

Consider the following execution timelines.

The notation in the timelines is as follows:

R\_addr(A) means the value A was read at memory address addr

W\_addr(A) means the value A was written to memory address addr

P1: W\_x(1)    W\_x(4)    R\_y(2)  W\_x(6)

P2:   W\_y(2)          R\_x(4)      W\_y(5)

P3:                                                                   R\_y(5)   R\_x(6)

P4:                                                                   R\_x(6)   R\_y(5)

P5:                                                                   R\_x(4)   R\_y(5)

P6:                                                                   R\_y(5)   R\_x(4)

P7:                                                                   R\_y(2)   R\_x(4)

P8:                                                                   R\_y(2)   R\_x(1)

Which one of the following is correct:

a)     (A) Processes P1, P2, P3 and P4 are causally consistent

b)    (B) Processes P1, P2, P5 and P6 are causally consistent

c)     (C) Processes P1, P2, P7, P8 are causally consistent

Select all that apply.

Answers: all are causally consistent. A, B – +1.5pt; C – +2pt

**Q13. [5 points]**

A datacenter currently has 100 servers, each with failure probability of 0.03. The datacenter manager is considering to upgrade the machines to newer, more powerful once, unfortunately with twice as large failure probability. If the manger wants to keep the probability of failure occurring anywhere in the datacenter to approximately same as it currently is, what should he do:

a)     double the number of machines

b)    half the number of machines

c)     keep the number of servers same as now

answer: half the number of machines.

you didn’t even need to do math. More components with same failure probability means more likely failure will occur somewhere. So it’s not (a). Here you also have that per machine failure probability doubled, so it’s not (c) either. => (b)

**Q14. [6 points]**

A distributed file system consisting of many file servers is implemented based on the following design decisions:

- The file system implements session semantics for non-concurrent writes

- The file system implements sequential semantic for concurrent writes

- Clients cache file data at block granularity

Considering the following possible choices, complete the sentences 1.-3. below, by writing in the letter corresponding to your selection in the provided blank space.

(A) only on open;

(B) only on close;

(C) only on open and close;

(D) on every file operation;

(E) every 30sec;

(F) on open, close, and every 30 sec;

1. Read clients contact a sever in the file system \_\_\_\_.

2. When there is a single writer in the system, the writer contacts a server in the file system \_\_\_\_.

3. When there are multiple writers in the system, the writers contact a server in the file system \_\_\_\_.

Answers:  [2pts each]

1) only on open (A)

2) on open and close (C)

3) on open, close and on every operation (D)

**Q15. [10 points]**

A distributed file system consisting of many file servers is implemented, just as in the previous question, based on the following design decisions:

- The file system implements session semantics for non-concurrent writes

- The file system implements sequential semantic for concurrent writes

- Clients cache file data at block granularity

Consider the following possible fields. Then, answer the questions below using comma-separated lists of the letters corresponding to the correct choices.

A. Timestamp (or version)

B. Current-writers

C. Dirty-blocks

D. Cacheable

E. Readers-caching

1. Which of these fields must appear in the client-side per-file data structures in order to support the implementation of the above design decisions? \_\_\_\_

2. Which of these fields must appear in the server-side per-file data structures in order to support the implementation of the above design decisions? \_\_\_\_

|  |
| --- |
|  |

Answers:  [1.5pts each correct letter + 1extra point]

1) must have: A, C, D

2) must have: A, B, D