

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name:_____CS 6210 Teaching Team_____GT Number:

Note:

1. **Write your name and GT number AT LEAST on the first page.**
2. The test is **CLOSED BOOK** and **NOTES**.
3. Please provide the answers in the space provided. You can use scratch paper (provided by us) to figure things out (if needed) but you get credit **only** for what you put down in the space provided for each answer.
4. For conceptual questions, **concise bullets (not wordy sentences)** are preferred. **YOU DON'T HAVE TIME TO WRITE WORDY SENTENCES...**
5. While it is NOT REQUIRED, where appropriate use figures to convey your points (a figure is worth a thousand words!)
6. **Illegible answers are wrong answers.**
7. **DON'T GET STUCK ON ANY SINGLE QUESTION...FIRST PASS: ANSWER QUESTIONS YOU CAN WITHOUT MUCH THINK TIME; SECOND PASS: DO THE REST.**

Good luck!

| Question number | Points earned | Running total |
|-------------------------------|---------------|---------------|
| 0 (0 minutes) (Max: 1 pts) | | |
| 1 (20 minutes) (Max: 33 pts) | | |
| 2 (20 minutes) (Max: 34 pts) | | |
| 3 (20 minutes) (Max: 32 pts) | | |
| Total (60 min) (Max: 100 pts) | | |

1. **(1 point, 0 min)** (This is a freebie, you get 1 point regardless)
OMSCS program was launched in
 - 2009
 - **2014**
 - 2018
 - 2019

OS Structures

2. **(33 points, 20 min)**
Imagine you are at the SOSP session wherein all three papers SPIN, Exokernel, and L3 microkernel are presented. Your friend and you are comparing notes on what you got out of each paper in terms of intellectual contributions that advances the state of the art in OS structuring.

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name: CS 6210 Teaching Team GT Number: _____

(i) (3 points) Your friend thinks SPIN has shown conclusively that an entire OS can be written in a high-level language that provides strong type safety. Would you agree with her point of view?

No, accessing some hardware resources (e.g., device registers in controllers) will require stepping outside the boundaries of Modula-3.

+1 No

+2 Strong reasoning as to why

+1 Reasoning given, but it's incomplete or not clear

(ii) (3 points) Upon page fault service by a library OS, the mapping <vpn, pfn> has to be installed into the TLB by Exokernel on behalf of the library OS. Therefore, your friend thinks that Exokernel cannot provide good performance compared to a monolithic design of an OS. How would you counter her argument?

Installing the mapping is a one-time cost which will get amortized over the lifetime of the process. Once the mapping is installed with the help of Exokernel, address translations during the running of the process will happen at hardware speeds.

Therefore, no loss of performance compared to a monolithic OS.

(+3) if the above sense (one time cost, amortized over the process run) is conveyed.

(-1) if one-time cost not mentioned

(-1) if once-installed translation at hardware speeds not mentioned.

(iii) L3 microkernel requires each subsystem to be in distinct architecture enforced protection domain. L3 is implemented on an

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name: _____ CS 6210 Teaching Team _____ GT Number: _____

architecture wherein there is no address-space tagging support in the TLB. The architecture does support segment registers.

(iii.a) (3 points) Your friend thinks that the performance is going to be terrible compared to SPIN due to the need for using architecture-enforced protection domains. What would be your counter argument?

SPIN packs logical protection domains within the same hardware address space.

For L3, the same advantage accrues by packing multiple protection domains (separated by hardware-enforced segment registers) within the same hardware address space.

So, no loss of performance in L3 compared to SPIN.

(+3) if the above sense is conveyed

(-2) if use of segment registers for separating logical protection domains not mentioned

(iii.b) (3 points) Your friend thinks that L3's approach to OS structuring would incur more implicit cost for protection domain switch compared to a monolithic design. Is she right or wrong? Defend your stand with justification.

Wrong. Implicit cost is purely dependent on the working set of a protection domain since the processor cache uses physical address tags.

(+3) if the above sense conveyed.

(-1) if physical address tag in processor cache not mentioned

(-1) if implicit cost dependence on working-set not mentioned

(a) (3 points)

Give one example of how SPIN's intellectual contribution can be traced to the state-of-the-art in modern systems.

SPIN introduced the concept of extensions. Any answer that talks about this (+3)

SPIN relied on language-level services like typechecking and dynamic linking, instead of relying on the hardware. Any answer that talks about this (+3)

(b) (3 points)

Give one example of how Exokernel's intellectual contribution can be traced to the state-of-the-art in modern systems.

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name: _____ CS 6210 Teaching Team _____ GT Number: _____

Virtualization using a hypervisor. Any answer that mentions this (+3)

(c) (3 points)

(Answer True/False with justification)

"SPIN and Exokernel are fair in comparing the superiority of their respective specialization approaches for OS services relative to Mach micro-kernel."

Mach focused on extensibility and portability; improving performance was not an objective. SPIN and Exokernel focused on improving extensibility and performance. So, the answer is No.

+3 for any answer that conveys the idea above.

(d) For this question assume a MIPS-style architecture wherein the hardware has ONLY a TLB for doing address translation (i.e., the architecture does not use a page table for address translation as is done in an architecture like x86). The architecture does provide instructions for entering VPN->PPN mappings and invalidating the mappings in the TLB. You have implemented a library OS on top of Exokernel. Assume you have obtained authorization from Exokernel to enter any VPN->PPN mappings into the TLB. Assume Exokernel provides an API for library OSes to enter VPN->PPN mappings into the TLB. Assume that Exokernel provides an API call to obtain a set of physical page frames at bootup time of a library OS.

(i) (4 points) Your OS is creating a new process. What are the steps your library OS will take to make the process runnable?

Example answer:

1. The OS checks if it has free page frames pre-allocated from Exokernel. Otherwise, it asks the Exokernel for authorization for free page frames. Exokernel creates secure binding for them and exposes them to the OS.
2. The OS allocates a PT data structure for the new process using one of its free page frames. (+1)
3. Depending on the OS policy, it allocates some free page frames for building the memory footprint for the process. Depending on the OS policy, it loads the some of the code, and global data pages of the process into the page frames allocated for the process (providing the necessary authentication to Exokernel). (+1)
4. The OS creates a VPN to PPN mapping for the process using the allocated page frames and marks them valid in the PT data structure. (+1)
5. The OS calls the API provided by the Exokernel to enter VPN->PPN mappings into the TLB, along with the authentication for the TLB. (+1)
6. The process is ready to be run.

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name: CS 6210 Teaching Team GT Number: _____

(ii) (4 points) The new process is running on the CPU. It makes a "malloc" system call to grow its heap space. How is this handled?

Example answer:

1. The "malloc" call gets trapped to Exokernel.
2. Exokernel looks up the event handlers associated with this library OS in the PE data structure.
3. Exokernel upcalls through the registered handler to the library OS.
4. The library OS serves this request (OS requests more free page frames from Exokernel if needed; OS enters new VPN to PPN mappings to TLB). Return control back to the process.

Rubric:

- +1 for mentioning syscall gets trapped to Exokernel
- +1 for mentioning Exokernel looks up PE for event handler
- +1 for mentioning Exokernel upcalls OS
- +1 for mentioning OS serves syscall

- (e) (2 points)
(Answer True/False with justification)
"SPIN is a microkernel"

False.

SPIN provides NO abstractions usually associated with a microkernel.

It simply provides header files for functions for each subsystem that form the core components of an OS, and an eventing mechanism to upcall into these functions from SPIN. The developer by populating these functions builds a monolithic OS. So, SPIN plus its extension is an OS not a microkernel.

- +0, Only False
- +1, False with incorrect justification
- +2, False with correct justification

- (f) (2 points)
(Answer True/False with justification)
"Exokernel is a microkernel"

False.

Exokernel DOES NOT provide any abstractions (e.g., threads, IPC, virtual memory) that a microkernel would.

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name: _____ CS 6210 Teaching Team _____ GT Number: _____

It simply provides a set of APIs to securely expose hardware resources at a fine granularity for building system services above the Exokernel. In this sense Exokernel is neither a microkernel nor a full-fledged OS.
+0, Only False
+1, False with incorrect justification
+2, False with correct justification

Virtualization

3. (34 points, 20 mins) (Paravirtualization)

You have implemented Xenolinux on top of Xen. You have implemented the network layer that does packet send/receive using two I/O rings: one for transmit and one for receive.

(a) (4 points) How will you ensure zero-copy semantics (i.e., no copying from your Xenolinux to Xen) for transmitting a packet?

[2 points] Guest OS enqueues descriptors in the I/O ring containing pointers in its address space to the transmit buffer containing the packets to be transmitted, which can then be directly DMAed by the NIC.

[2 points] Pages corresponding to the above transmit buffers are pinned for the duration of the transfer.

(b) (4 points) How will you ensure zero-copy semantics for receiving a packet from Xen into Xenolinux?

[2 points] Guest OS pre-allocates pages for receive buffers so that an incoming packet can be DMAed directly into the receive buffer.

AND/OR

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name: _____ CS 6210 Teaching Team _____ GT Number: _____

[2 points] Xen DMA's the incoming packet into its own buffer; it then exchanges that page frame for one of the page frames owned by the Guest OS, and stores the descriptor into the I/O ring shared with that Guest OS.

(c) (4 points) Multiple processes on top of Xinolinux wish to transmit at the same time. How do you handle this situation in your implementation?

Ordering requests from multiple processes by the Guest OS is orthogonal to the I/O ring mechanism used for communicating between a given Guest OS and Xen. So far as Guest OS and Xen is concerned, the requests are enqueued one at a time into the I/O ring.

(+4) if the above sense conveyed.

(may have to decide on what partial credit depending on the answers).

(d) (8 points) (Full Virtualization)

Assume a guest-OS has started 4 processes in a fully virtualized environment on a 32-bit machine. Assuming 4K page size, explain how many entries this guest-OS has in the shadow page table.

(e) (4 points) (Full Virtualization)

Assume an architecture which uses a page table for address translation. The CPU has a PTBR to point to the current page table used by the processor for address translation.

A process P1 is executing on top of a fully virtualized OS. The OS wishes to context switch from P1 to P2.

List the steps before P2 starts execution on the processor.

1. Guest OS executes the privileged instruction for changing the PTBR to point to P2's page table. Hypervisor traps this execution. (+1)
2. From the PPN of the PT for P2, the hypervisor will know the offset into the S-PT for that guest-OS where the PT resides in machine memory. (+2)
3. Hypervisor sets the PTBR to the MPN thus located as the PT for P2. (+1)

-1 pts

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name: _____ CS 6210 Teaching Team _____ GT Number: _____

No mention of Guest OS executing a privileged instruction for changing the PTBR to point to P2's page table.

-1 pts

No mention of instruction trapping into the hypervisor.

-1 pts

Partial explanation of how the hypervisor will locate the MPN of PT for P2

-2 pts

No mention or incorrect explanation of, how the hypervisor will locate the PPN of PT for P2

-1 pts

No mention of Hypervisor sets the PTBR to the MPN thus located as the PT for P2.

-0.5 pts

Partial mention of Hypervisor sets the PTBR to the MPN thus located as the PT for P2.

(f) (10 points) (Memory management)

Assume all the VMs running in a data center have balloon drivers installed.

VM4 experiences memory pressure. It asks the hypervisor for 350 MB additional memory.

The state of the other VMs are as follows:

- VM1 has 100 MB of machine memory it is not actively using.
- VM2 has 500 MB of machine memory it is not actively using.
- VM3 has 300 MB of machine memory it is not actively using.

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name: _____ CS 6210 Teaching Team _____ GT Number: _____

For each allocation request, the policy of the hypervisor for memory management is to exercise a 20% TAXATION on idle memory iterating through ALL the VMs, starting with the VM that has the most idle memory until the required allocation request is met. The hypervisor may iterate through the VMs with idle memory (multiple times if need be) if the allocation request is not met in one pass through all the VMs.

List the steps by which VM4's memory pressure is alleviated. Your answer should quantify the amount of memory taken from each VM to relieve VM4's memory pressure.

20 MB from VM1, 80MB remaining in VM1, 330MB ask remaining
100MB from VM2, 400MB remaining in VM2, 230MB ask remaining
60MB from VM3, 240MB remaining in VM3, 170MB ask remaining
16MB from VM1, 64MB remaining in VM1, 154MB ask remaining
80MB from VM2, 320MB remaining in VM2, 74MB ask remaining
48MB from VM3, 192MB remaining in VM3, 26MB ask remaining
12.8MB from VM1, 51.2MB remaining in VM1, 13.2MB ask remaining
64MB from VM2, 256MB remaining in VM2, 0 ask remaining
(or)
13.2MB from VM2, 306.8MB remaining in VM2, 0 ask remaining

2.5 pts

Mentioned ballon driver's involvement in the operation

+ 2.5 pts

Applied 20% tax to VM2 first

+ 2.5 pts

Applied 20% tax to VM3 second

+ 2.5 pts

Applied 20% tax to VM1 third, up to the fulfillment of the request

- 1 pts

Took away more memory than necessary

- 3 pts

Did not follow the algorithm's order

Parallel Systems

4. (32 points, 20 mins)

(a) (4 points) (atomicity)

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name: _____ CS 6210 Teaching Team _____ GT Number: _____

You are implementing an invalidation-based cache coherent shared memory multiprocessor, wherein each processor has a private cache. You start with a uniprocessor as a basic building block. The ISA of the processor supports an atomic Test-and-Set (T&S) instruction. Your aim is to make sure that the T&S operation is globally atomic. What design choices are available to you to achieve this aim?

Answer

1. Execute the RMW operating in the cache controller: Use the invalidation-based CC protocol to obtain exclusive ownership for the memory location on which T&S is being performed and execute the RMW operation
2. Extended ownership of the interconnect: Give exclusive ownership of the interconnect (e.g., bus) so that the processor can complete the T&S operation to the memory location bypassing the cache.
3. Execute the RMW in the memory controller: Bypass the cache and send the T&S request to the memory controller. The memory controller serializes the T&S operation from different processors to the same memory location.

(+4 if at least 2 of the above three design choices are mentioned in some fashion; don't be anal about the exact wording)

(b) (4 points) (memory model)

Your co-worker wants to provide a sequential consistency memory model to the application programmer on top of your multiprocessor. How can you take care of her requirement in your cache coherent multiprocessor design?

Answer

- Sequential consistency can be ensured by using cache coherence protocols to ensure exclusive access to a memory location before writing to it.
- For e.g. If we use an invalidate based coherence protocol, all cached copies of the cache line will be invalidated before a processor writes to that cache line. Other processors can fetch the updated cache line after the write (from the processor or memory).

(+4 if the above sense is conveyed)

(c) (4 points) (spinlock)

Consider the following lock algorithm using T&S:

```
while ((L == locked) or (T&S(L) == locked))
{
    while (L == locked); // spin
}
```

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name: _____ CS 6210 Teaching Team _____ GT Number: _____

```
    delay (d[Pi]); // different delays for different processors
}
// success if we are here
```

(i) (Answer True/False with justification) (No credit without justification)
This algorithm does not rely on hardware cache coherence.

False.

Processors spin on cached value (While (L==locked)). The only way for them to come out of the spin loop is if the cached value changes which necessitates hardware cache coherence.

+0, Only False

+1, False with incorrect justification

+2, False with correct justification

(ii) (Answer True/False with justification) (No credit without justification)

The algorithm performs especially well under high lock contention.

True.

Upon lock release, different processors wait for different amount of delay times thus reducing the contention on the bus.

+0, Only True

+1, True with incorrect justification

+2, True with correct justification

(d) (4 points) (spinlock)

The ticket lock algorithm shown below gives fairness and each spinning processor spins for a different amount of time commensurate with its expected wait time for the lock before testing if the lock is available.

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)


Name: _____ CS 6210 Teaching Team _____ GT Number: _____

```
struct lock{
    int next-ticket;
} int now-serving;

acquire-lock(L):
    int my-ticket = fetch-and-inc(L->next-ticket);

    loop
        pause (my-ticket - L->now-serving);
    if (L->now-serving == my-ticket) return;

release-lock(L):
    L->now-serving++;
```



What (if any) are the reasons for this algorithm to not work well?

(For an update-based CC protocol):

Upon each lock release $L \rightarrow \text{now_serving}$ is updated \Rightarrow unnecessary bus transaction since the change is meaningful for only ONE processor that is next in line to get the lock.

(For invalidation-based protocol):

The duration of pause (spin loop) is just a guestimate of the expected wait time for a processor. Every time it comes out to check if it is its turn for the lock, there would be bus transaction to get $L \rightarrow \text{now_serving}$ if its cached value has been invalidated (due to an unlock operation).

(+4 if either reason is given)

(OR +2 if contention mentioned but no valid reason given)

- (e) (6 points) (barrier)
(Answer True/False with justification. Zero credit without justification.)
(i) "MCS barrier will not work on a NCC-NUMA architecture."

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name:_____CS 6210 Teaching Team_____GT Number:

False.

The parent will be spinning on a memory location in its NUMA piece of the memory. The child will reach across the ICN to modify the memory location ear-marked for it. The change will be seen by the parent since there is hardware cache coherence WITHIN each node of the NCC-NUMA architecture.

+0, Only False

+1, False with incorrect or vague justification

+2, False with correct justification (if the above sense is conveyed without being anal about the wording)

(ii) "The total communication complexity of dissemination barrier is $O(N \log_2 N)$ "

True.

Every round will have one message sent by every node $O(N)$ [1].

There are $O(\log_2(N))$ rounds [2]. Hence communication complexity is $O(N \log_2(N))$.

+0, Only True

+1, True with incorrect or vague justification

+2, True with correct justification (if the above sense is conveyed without being anal about the wording)

(iii) "The tournament barrier works with both shared memory and message-passing (i.e., clusters) architectures."

True.

The communication between nodes can be either messages sent from one node to another, or shared memory location where one node is spinning locally and the other writes.

+0, Only True

+1, True with incorrect or vague justification

+2, True with correct justification (if the above sense is conveyed without being anal about the wording)

(f) (2 points) (communication)

In RPC, succinctly state the role played by the client-stub and the server-stub.

Client stub: The actual arguments of the call are serialized into a contiguous RPC message packet to pass to the kernel.

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name: _____ CS 6210 Teaching Team _____ GT Number: _____

Server stub: Takes the contiguous RPC message received from the kernel and de-serializes it to populate the server procedure's stack with the actual arguments of the call.

+1 if serialization of arguments mentioned for client stub

+1 if deserializing the message into arguments mentioned for server stub

(g) (4 points) (scheduling)

A multi-threaded multicore CPU is one in which each chip has multiple cores and each core has multiple hardware threads. The OS chooses the set of application threads to be scheduled on the hardware threads in each core. Given that the hardware threads share a single processor pipeline on the core,

(i) What purpose is served by the hardware threads?

The purpose of multiple hardware threads in each core is to ensure efficient utilization of the single processor pipeline available to each core. If a hardware thread has to perform a long latency operation, e.g., miss in the last level cache (LLC) causing the processor to go off-chip to fetch the missing access from the memory (which can take 100 or more CPU cycles), the hardware can switch to another hardware thread.

(+2 all or nothing)

(ii) What should the OS do ensure that processor pipeline is utilized well? Why?

In scheduling the set of threads to run on the chip, the OS should ensure that the combined working sets of all the threads (number of cores X number of hardware threads per core) can be packed into the last level cache to reduce the occurrence of long latency operations (i.e., misses in the LLC).

(+2 all or nothing)

(h) (4 points) (memory manager for multiprocessor)

You are implementing the virtual memory manager for your multiprocessor OS. You have a page fault handler that executes independently in each processor. If there is a page fault for the currently executing thread/process, then the handler on that processor deals with it without disturbing the activities on the other processors. Your OS supports both single-threaded processes as well as multi-threaded processes. You implement your memory management system in the conventional manner with a page table per process that provides the mapping of the VPN to PPN (or the disk address if it is not in physical memory).

CS 6210 Fall 2022 Test1 (120 min Canvas Quiz)

Name: _____ CS 6210 Teaching Team _____ GT Number: _____

(i) Does your design ensure that if there are concurrent page faults incurred by independent processes running on different processors, they will be handled by your memory manager concurrently? Justify your answer.

Answer - Yes. Since we have a page table per process, the page table needed to cater to different page faults would be different.

(all or nothing if the above sense is conveyed)

(ii) Does your design ensure that if there are concurrent page faults incurred by threads of the same process running on different processors, they will be handled by your memory manager concurrently? Justify your answer.

Answer - No. Since the mappings from VPN to PPN for different threads of the same process will be present in the same page table, the memory manager would need to serialize the handling of these page faults.

(all or nothing if the above sense is conveyed)