CS 6210 Fall 2010 Midterm

Name:________Kishore________________________GT Number:

Thursday October 21, 2010 (4:00 to 5:30 PM)

Note:

1. **Write your name and GT number on each 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.
5. Where appropriate use figures to convey your points (a figure is worth a thousand words!)
6. Illegible answers are wrong answers.
7. Please look through the whole test before starting so that you can manage your time better.

Good luck!

Question number	Points earned	Running total
1 (0 min) (Max: 1 pts)		
2 (20 min) (Max: 20 pts)		
3 (10 min) (Max: 10 pts)		
4 (10 min) (Max: 9 pts)		
5 (10 min) (Max: 10 pts)		
6 (10 min) (Max: 10 pts)		
7 (10 min) (Max: 15 pts)		
8 (10 min) (Max: 10 pts)		
9 (10 min) (Max: 15 pts)		
Total (90 min) (Max: 100 pts)		

1. (1 min, 1 point)
The capital of Silla kingdom
(Don’t worry you get a point irrespective of your answer!)

a. Seoul
b. Pyongyang
[✓ c. Gyeongju]
d. Busan
e. Gumi
f. None of the above

CS 6210 Fall 2010 Midterm

Name:_________Kishore___________________________GT Number:

2. (20 min, 20 points) (OS Structures - conceptual)

(a) (4 points) Choose the correct **two** choices pertaining to **SPIN** from the following (**-2 for each incorrect choice**)

A. SPIN keeps each logical protection domain in a distinct address space.

B. SPIN does not incur border crossing overhead for protection domains that have been collocated with the kernel through kernel extension.

C. Capabilities to entry points in protection domains are implemented with encrypted keys in SPIN.

D. In SPIN, hardware events (such as a page fault) result in the direct execution of interface procedures that have been registered by a protection domain with the kernel through the extension model.

(b) (4 points) Choose the correct **two** choices pertaining to **Exokernel** from the following (**-2 for each incorrect choice**)

A. Once acquired through secure binding, a library OS can hold on to resources as long as it wants.

B. Processor environment (PE) data structure in Exokernel contains, among other things, information pertaining to entry points in the library OS for upcalls from the Exokernel.

C. Upon a TLB miss, Exokernel **always** calls the faulting library OS for servicing the miss.

D. The mechanism in Exokernel for downloading code into the kernel is to avoid border crossing for protection domains that are critical for achieving high performance.

(c) (4 points) Choose the correct **two** choices pertaining to processor architecture and address spaces (**-2 for each incorrect choice**)

A. Implementation of protection domains **always** requires switching processor address space.

B. Translation Lookaside Buffer (TLB) is **always** flushed upon an address space switch.

C. Segmentation in hardware allows independent logical protection domains to share the same page table.

D. Keeping very large protection domains in distinct address spaces is a reasonable choice since indirect costs (i.e., reloading the TLB and cache effects) dominate the overall penalty for border crossing into such large protection domains.

(d) (4 points) (**Answer True/False with justification**) In implementing a high-performance microkernel, one should keep the implementation of the microkernel abstractions to be architecture independent.

False. As Liedke argues in his paper on L3 microkernel, to achieve high performance the microkernel implementation should fully exploit the hardware features available in the processor architecture.

CS 6210 Fall 2010 Midterm

Name:________Kishore________________________GT Number:

(e) (4 points) Choose the correct **two** choices pertaining to the paper "**Xen** and the art of virtualization" (**-2 for each incorrect choice**)

A. System calls executed by a user process use a table lookup inside Xen to do an upcall to the appropriate guest operating system.
B. Implementation of Xen on Intel x86 architecture requires special handling of page faults incurred by a user process before the page fault can be handed over to the guest operating system.
C. Xen allows unmodified binaries of guest operating systems to be run on top of it.
D. Xen allows fine grained access to hardware resources by a guest operating system exactly similar to Exokernel.

3. (10 min, 10 points) (Exokernel - conceptual)
Consider a user process executing in a library OS on top of the Aegis Exokernel. The user process incurs a page fault (i.e., a TLB miss). Explain the action that follows in each of the following situations. You should identify the mechanisms and data structures available in Exokernel that enable it to accomplish the action.

a) (2 points) The faulting virtual address is a guaranteed mapping for the library OS.

- using the PE data structure for this library OS, Exokernel looks up its software TLB to get VPN to PFN mapping for this VA
- Exokernel installs this mapping into the TLB + resumes execution of the user process

b) (4 points) The faulting virtual address is not a guaranteed mapping for the library OS but a valid mapping exists for the virtual page in the page table of the faulting process.

- using the PE data structure for this library OS, Exokernel determines the Exception context and upcalls the library OS
- library OS looks up the page table for this process, and retrieves the mapping (VPN, PFN) corresponding to this process
- library OS calls TLB update primitive available in Exokernel to put this mapping into the TLB by presenting the capability it has for the PFN
- Exokernel resumes execution of faulting process

CS 6210 Fall 2010 Midterm

Name:________Kishore________________________GT Number:

c) (4 points) This is the first access to the faulting address by the user process.

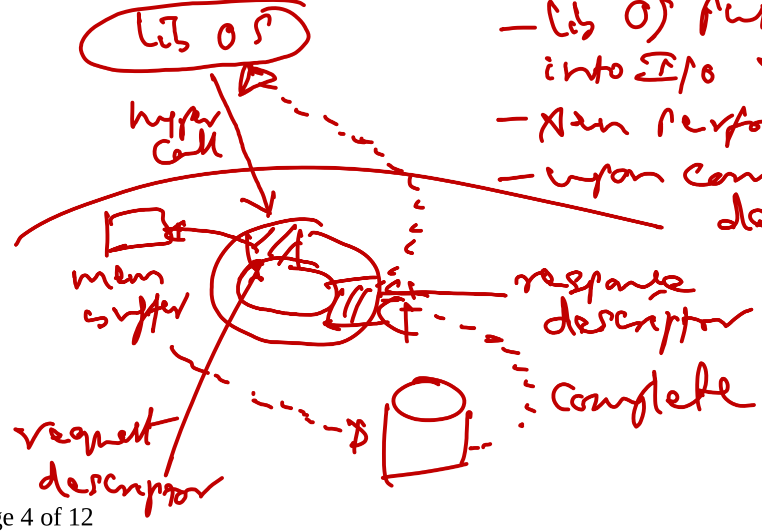
- As in the answer to part (b), Exokernel calls the entry point in the library OS for handling page faults (using the PE data structure)
- Library OS calls its hard disk driver to initiate I/O from disk to a physical page frame
- Device driver presents the capability for the hard disk controller + the page frame to Exokernel to perform the DMA
- Once DMA complete Exokernel uses PE (interrupt context) to upcall library OS
- As in answer to part (b), library OS completes TLB update

4. (10 mins, 9 points)(Xen – conceptual)

(a) (5 points) Using an example, describe the role performed by I/O rings in the Xen architecture. Use pictures to make your explanation easy to follow.

- used for data transfer and command indication between library OS + Xen

Example: Disk write



- Lib OS puts request descriptor into I/O ring
- Xen performs disk write
- upon completion Xen puts response descriptor for Lib OS in I/O ring

CS 6210 Fall 2010 Midterm

Name:_________Kishore___________________________GT Number:

(b) (4 points) Explain the concept and use of fastpath handlers in Xen.

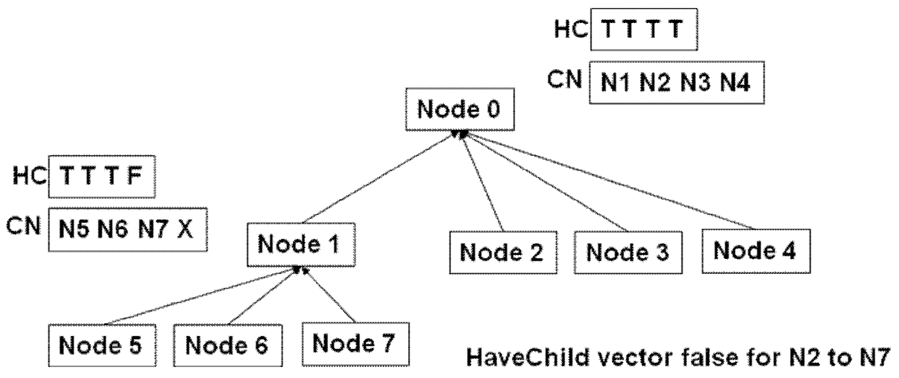
Concept:

- allows processes on a library OS to access entry points in the library OS without going through Xen
- implemented by Xen keeping a table for each library OS of traps and corresponding entry points

Use:

- system calls implemented using fastpath handlers
- registered once by library OS at startup
- upon system call, transfer control to entrypoint via table lookup

5. (10 mins, 10 points)(MCS tree barrier algorithm – conceptual)
Consider the following figure that shows the arrival tree for an 8-processor version of the MCS Tree barrier algorithm.



HC stands for "havechild"; CN stands for "childnotready"; the up arrow from each child represents the "parentpointer" to a child's specific location in the CN data structure of the parent.

Just to refresh your memory the MCS tree barrier algorithm is given on the next page:

CS 6210 Fall 2010 Midterm

Name:________Kishore________________________GT Number:

```
type treenode = record
    parentsense : Boolean
    parentpointer : ^Boolean
    childpointers : array [0..1] of ^Boolean
    havechild : array [0..3] of Boolean
    childnotready : array [0..3] of Boolean
    dummy : Boolean     // pseudo-data

shared nodes : array [0..P-1] of treenode
    // nodes[vpid] is allocated in shared memory
    // locally accessible to processor vpid
processor private vpid : integer     // a unique virtual processor index
processor private sense : Boolean

// on processor i, sense is initially true
// in nodes[i]:
//    havechild[j] = true if 4*i+j < P; otherwise false
//    parentpointer = &nodes[floor((i-1)/4)].childnotready[(i-1) mod 4],
//       or &dummy if i = 0
//    childpointers[0] = &nodes[2*i+1].parentsense, or &dummy if 2*i+1 >= P
//    childpointers[1] = &nodes[2*i+2].parentsense, or &dummy if 2*i+2 >= P
//    initially childnotready = havechild and parentsense = false

procedure tree_barrier
    with nodes[vpid] do
        repeat until childnotready = {false, false, false, false}
        childnotready := havechild    // prepare for next barrier
        parentpointer^ := false       // let parent know I'm ready
        // if not root, wait until my parent signals wakeup
        if vpid != 0
            repeat until parentsense = sense
        // signal children in wakeup tree
        childpointers[0]^ := sense
        childpointers[1]^ := sense
        sense := not sense
```

Explain the sequence of actions by which the algorithm determines that all the nodes have arrived at the barrier.

with reference to the arrival tree:

1) parent at each level waits for its children

2) children at leaf nodes use their unique location in the nodes [parent] using parentpointer to announce it has arrived [e.g. N2-N7]

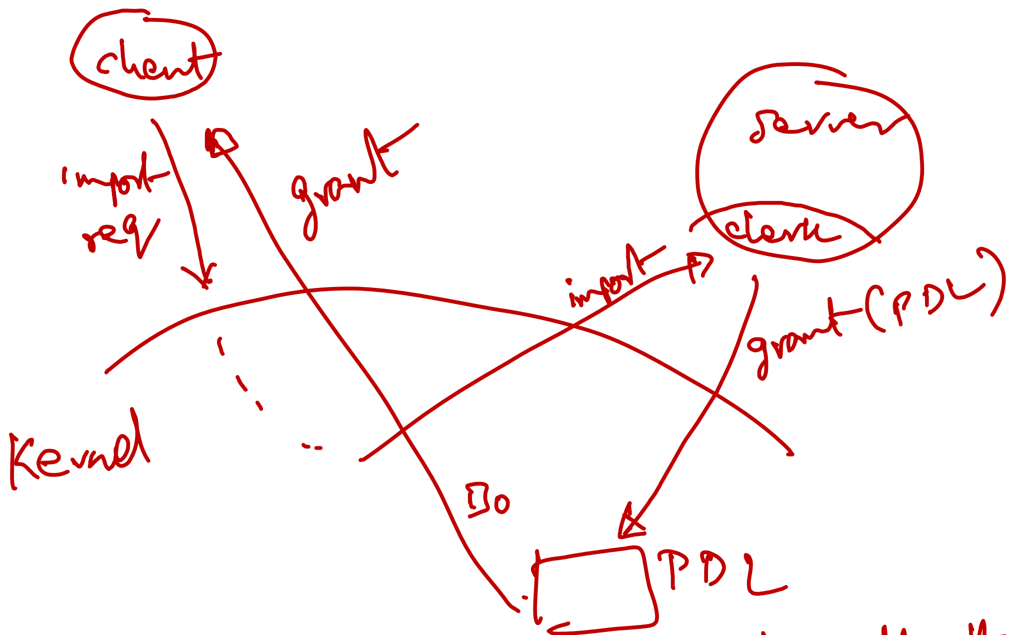
3) an intermediate node (e.g. N1) that arrives at barrier waits for all its children to arrive at the barrier by spinning on childnotready field of its nodes[] entry, and then does step 2

4) root node (e.g. N∅) when it arrives at a barrier waits for its children; once they have arrived algorithm recognizes that all nodes have arrived

CS 6210 Fall 2010 Midterm

Name:________Kishore________________________GT Number:

6. (10 mins, 10 points) (LRPC – conceptual)
Enumerate the actions (using figures) in LRPC in response to an import call from a client.



- upon import call, the kernel calls the clerk in the server with the details of import call
- clerk determines the entry point address and return details (entry point, number of args, number of results, etc) to the kernel in PDL
- kernel uses PDL to determine sizes of A-stack + E-stack; allocates A-stack and maps it into client-server address spaces
- kernel returns BO to client for future calls from this client to this specific RPC entry point in server
- client presents BO for future calls

CS 6210 Fall 2010 Midterm

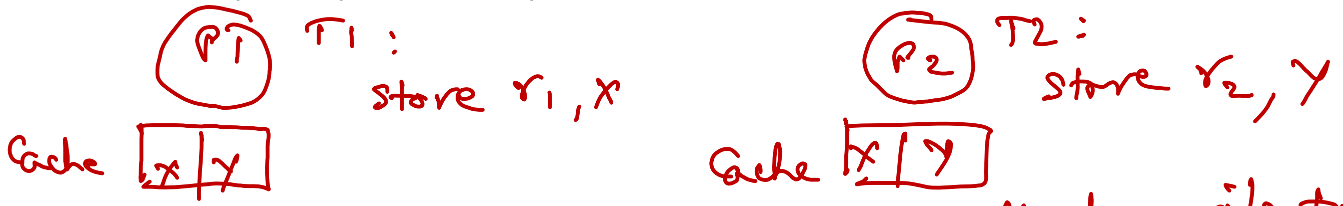
Name:________Kishore________________________GT Number:

7. (15 mins, 15 points) (Parallel systems – conceptual)

(a) (5 points) Consider an operating system for a shared memory multiprocessor. The operating system is structured to execute independently on each processor to handle system calls and page faults occurring for processes and threads executing on that processor. Each process executes within an address space. Multiple threads of the same process execute within the address space of the process. The operating system uses a single page table in shared memory for each process. Explain the performance problem with this approach if the application process is multithreaded.

Simultaneous page faults from different threads to different parts of the process address space cannot be serviced concurrently by the OS despite the availability of hardware parallelism.

(b) (5 points) Explain false sharing using a simple example (use pictures to explain your answer).



write to x by T1 does not affect write to y by T2 ; yet since x + y are in same cache line, there will be overhead of cache coherence actions. This is false sharing

CS 6210 Fall 2010 Midterm

Name:________Kishore________________________GT Number:

(c) (5 points) Consider the following situation in a multiprocessor.

- The history of execution on Processor P1 is T1, T2, T1, T3, T2
- The history of execution on Processor P2 is T3, T2, T4, T3
- The history of execution on Processor P3 is T1, T2, T3, T4, T5

In the above history, time flows from left to right. That is T2 is the most recent thread to execute on P1; T3 is the most recent thread to execute on P2; and T5 is the most recent thread to execute on P3. Thread T1 is ready to run again and can be scheduled on any of the above three processors. Which processor would you recommend to use for scheduling T1 and why?

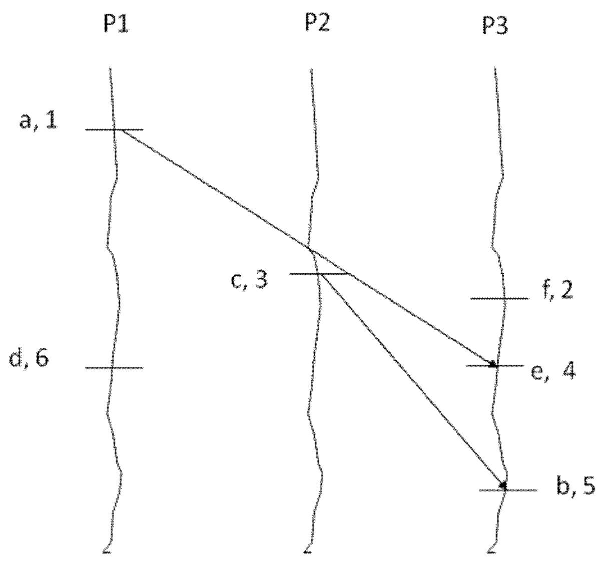
T1 was never run on P2 => cache on P2 has no content of T1

Four other tasks have run on P3 since T1 ran on it

Only two other tasks have run on P1 since T1 ran on it

Therefore, I would choose P1 to run T1 since P1's cache is likely less polluted

8. (10 mins, 10 points) (Distributed systems)
(a) (5 points) (Lamport's clock – problem)



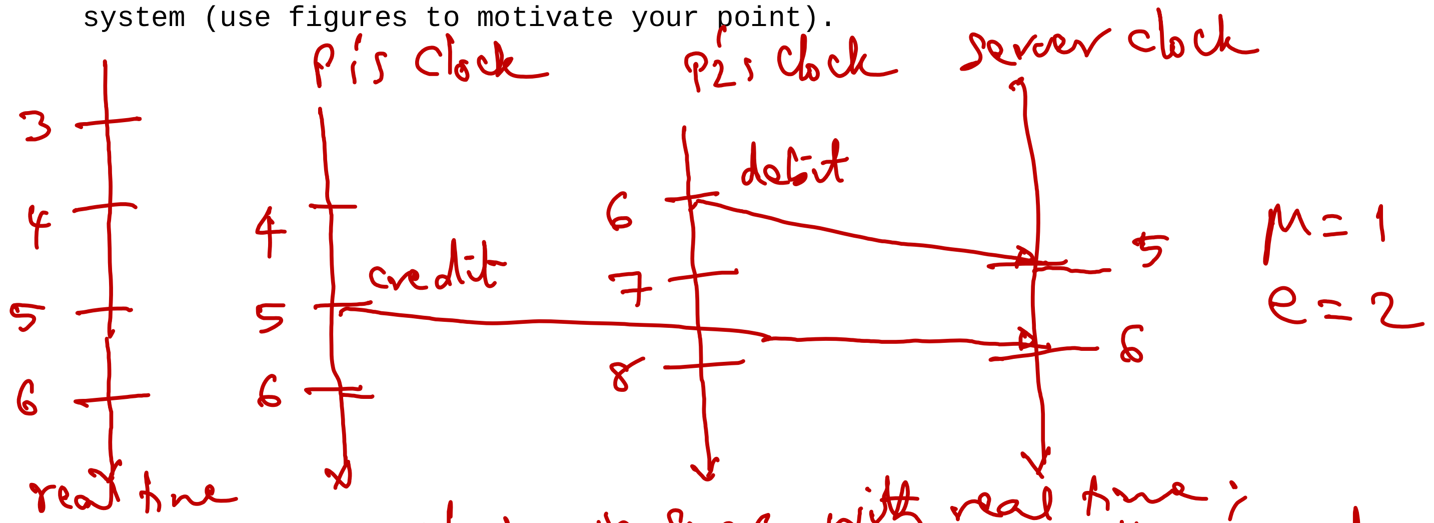
CS 6210 Fall 2010 Midterm

Name:________Kishore________________________GT Number:

Choose -> or || to show the relationship between the following events:

(a, 1)	$\longrightarrow$	(e, 4)
(a, 1)	||	(c, 3)
(a, 1)	$\longrightarrow$	(d, 6)
(f, 2)	||	(d, 6)
(a, 1)	$\longrightarrow$	(b, 5)

(b) (5 points)
With a simple example motivate the need for physical clocks in a distributed system (use figures to motivate your point).



P1 + Server clocks in sync with real time; P1 tells P2 out
mutual drift between P1 + P2 = 2; Due to mutual drift P2's
of band to debit after 5pm;
request arrives at server earlier in real time
+ hence rejected

9. (15 mins, 15 points) (Distributed systems)
(a) (5 points) (Thekketh and Levy – conceptual)
Identify the components of an RPC call that **ARE in the critical path of the latency** (**Choose all that apply; -1 for each incorrect choice**):

A. Time on the wire
B. Controller latency
C. Switching client out at the point of call
D. Interrupt service
E. Switching server in on call arrival
F. Switching server out on call completion
G. Switching client in to receive results of the call

CS 6210 Fall 2010 Midterm

Name:________Kishore________________________GT Number:

(b) (5 points) (Active networks – conceptual)
The authors identify three sources of protection concerns in using active networks: (i) corruption of ANTS runtime at an active node, (ii) code spoofing in the packets, and (iii) corruption of the soft state at an active node. Explain how ANTS toolkit addresses each of these protection threats.

(i) Java Sandboxing

(ii) Capsule type is encrypted; Spoofing Possible only if code can be cracked

(iii) restricted API; soft-state also protected by the same encrypted cipher limiting extent of access to soft-state for each packet

CS 6210 Fall 2010 Midterm

Name:________Kishore_________________________GT Number:

(c) (5 points) (Ensemble/NuPrl)
Develop an abstract specification and a concrete specification for Lamport's distributed mutual exclusion algorithm. You do not have to adhere to any specific syntax (such as IOA), and can invent your own so long as it is straightforward and understandable.

Abstract Spec :

<u>variable</u>

Q: Global Queue of requests

<u>Actions</u>

request :

<u>enqueue</u> request in Q at <u>tail</u>

if my request at head enter CS

release :

<u>remove</u> my request from Q

Concrete Spec :

<u>variables</u>

Q : private per processor Queue of requests

C : private per processor clocks init to ∅

<u>Actions</u>

request_send :

<u>enqueue</u> my request in Q at tail;

<u>send</u> req to all peers; increment C;

<u>receive</u> acks from all peers

if my request at head enter CS

request_receive :

enqueue received request at appropriate place in Q; send ACK; increment C;

release_send :

dequeue my request from Q; send release to all peers

release_receive :

dequeue request from Q