Exams

GLOBAL DOUBT:
Pointers allowed in RPC ?

Design	Communication	Evaluation
THE	Ring system	
Nucleus	Message Passing (mail boxes)	
Opal	Shared Memory	
Exokernel	Message passing	
Multikernel	Message passing	
Disco	Shared Memory	
VMWare	Shared Memory	
Mach	Message passing + Heavy CoW for sharing	
SuperPage		
RPC	RPC Calls	
LRPC	Message based use A stack	
Sche Act	Message Passing (otherwise slow)	
Lottery	N/A	
COOP	Message based	Farsite

SWIFT PAPER

1. Difference between policy and mechanism
Policy - what you want to do,
Mechanism - how you want to do.

E.g., CS537 - policy to maintain the top 20 commands in history. Mech: Using array / linked list.

VMWare

Policy	Mechanism
Do not change the OS to revoke memory	Ballooning.
Content based page sharing	Use the hash list to compare - so that you don't make a CoW for all the pages. And use the hint list to see if there was a change occurred.
Tax based allocation of resources	Tax a VM which is using less number of

	resources actively and based on it, a VM has a higher chance of being penalized. Active working set was computed by invalidating TLB entries and then statistically checking how many were accessed.

Unix

Process - time slice based.	Use timer interrupts to implement the CPU scheduling.
File System - uniquely name and persistent	Inode structure and free block list.
Directory	Special files accessible only through specific APi cannot be modified as a regular file
Device / Regular Files	Implemented in the same way as files.

Opal

Single shared address space	
Sharing memory	Capability - map the region of memory into your address space.
Capabilities for inter domain system calls	Implemented through portals
Divide memory in a continuous address space	Protection domains - that define the set of virtual addresses that are valid for each thread.
Decoupling of protection domain from resource ownership	through resource groups allowing threads to switch RG or to create sub groups to run untrusted procedures

Capability - portal - it mechanism and policy. nth level is policy, to implement it, (n-1)th level becomes mechanism. Ask Swift

Q3.
A.

Working set - keep a process in RAM only if it all the pages it uses at a particular instance in time can be kept in RAM.
- Mark the entries corresponding to a process (using ASID) as invalid.
- Pages access in that time frame (TLB miss) will the be the considered as the reference count.
- This is a fair estimate of working set (same as what is described in VMWare ESX Server).

B.
Security in OS / Opal
OS - provides a virtually isolated addressing which ensures one process can't access another process's VM.
It is tied in to the concept of memory.

Opal uses protection domain
- which are the set of virtual address
- may lead to security bugs

MISCELLANEOUS CONCEPTS
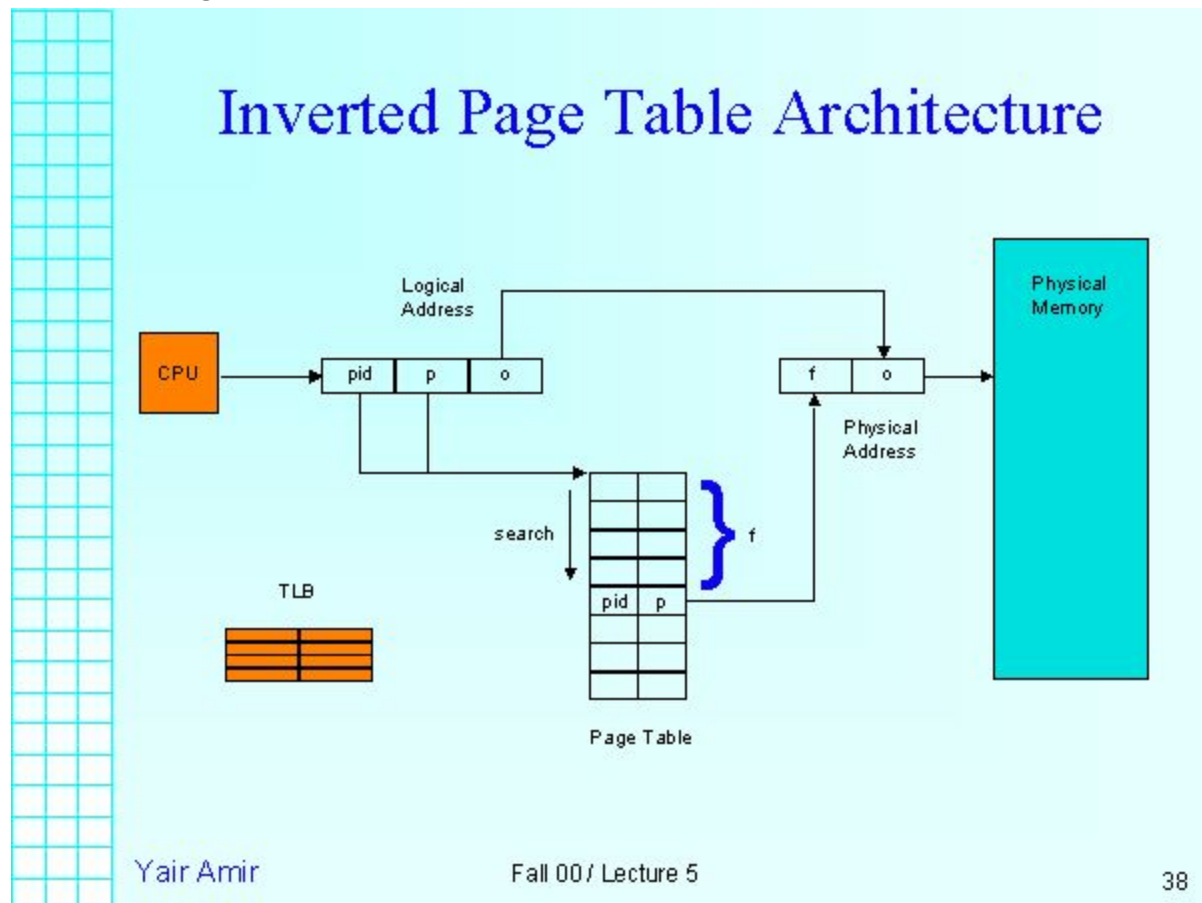**1. Opal eval**

2. Cache coloring
In computer science, cache coloring (also known as page coloring) is the process of attempting to allocate free pages that are contiguous from the CPU cache's point of view, in order to maximize the total number of pages cached by the processor. Cache coloring is typically employed by low-level dynamic memory allocation code in the operating system, when mapping virtual memory to physical memory. A virtual memory subsystem that lacks cache coloring is less deterministic with regards to cache performance, as differences in page allocation from one program run to the next can lead to large differences in program performance.

3. RADIX
In computer science, a radix tree (also radix trie or compact prefix tree) is a data structure that represents a space-optimized trie in which each node that is the only child is merged with its parent. The result is that the number of children of every internal node is at least the radix r of the radix trie, where r is a positive integer and a power x of 2, having x ≥ 1. Unlike in regular tries, edges can be labeled with sequences of elements as well as single elements. This makes radix trees much more efficient for small sets (especially if the strings are long) and for sets of strings that share long prefixes.
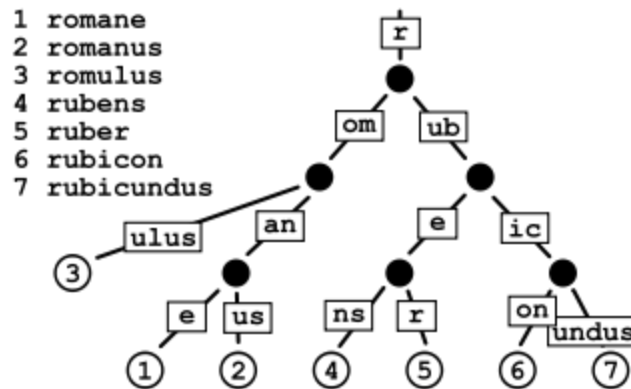
4. Inverted Page Table

Fully Associative TLB