Lecture 23 17th Sept. 2018

10 September 2018 19:38

— synchronization, process setup, context switch, scheduling, file systems | network IO virtualization (VMs)

situations :

- two processes call fork()
- processes on two cores to be moved to ready queue
- remove page from free list to service page fault.
- dequeue/enque pkts in pkt-buffer

fork { } P_1 P_2

fork() fork()

CPU1 CPU2

|| race condition

more than one execution entity accesses a memory region (shared) simultaneously.

|| variables objects

~ ready from mem
modify
update in mem.

scenarios in the Kernel.

(i) system call + system call

(ii) system call + Kernel thread

(iii) interrupts + interrupts

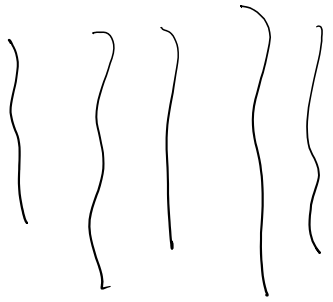
(iv) system call + interrupts

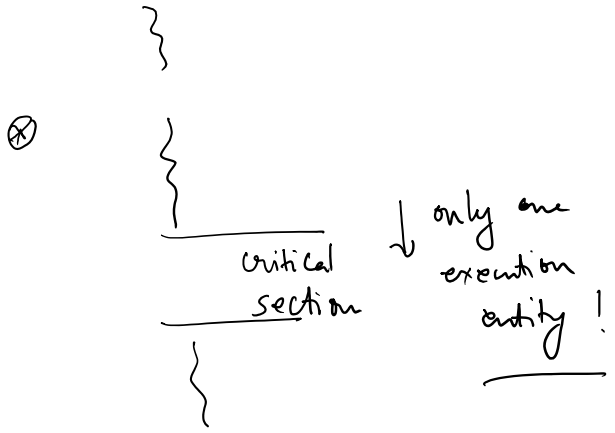
Q

|| is this only a multi-CPU problem?

```
insert (int data) {
    ... = head list;
```

```
insert ( int data ) {
1.      list * L = new list;
2.      L -> data = data;
3.      L -> next = head;
4.      head = L;
}
```





synchronization, co-ordination

locks atomic

mutex

semaphores

Condition variables

spin locks

deadlocks