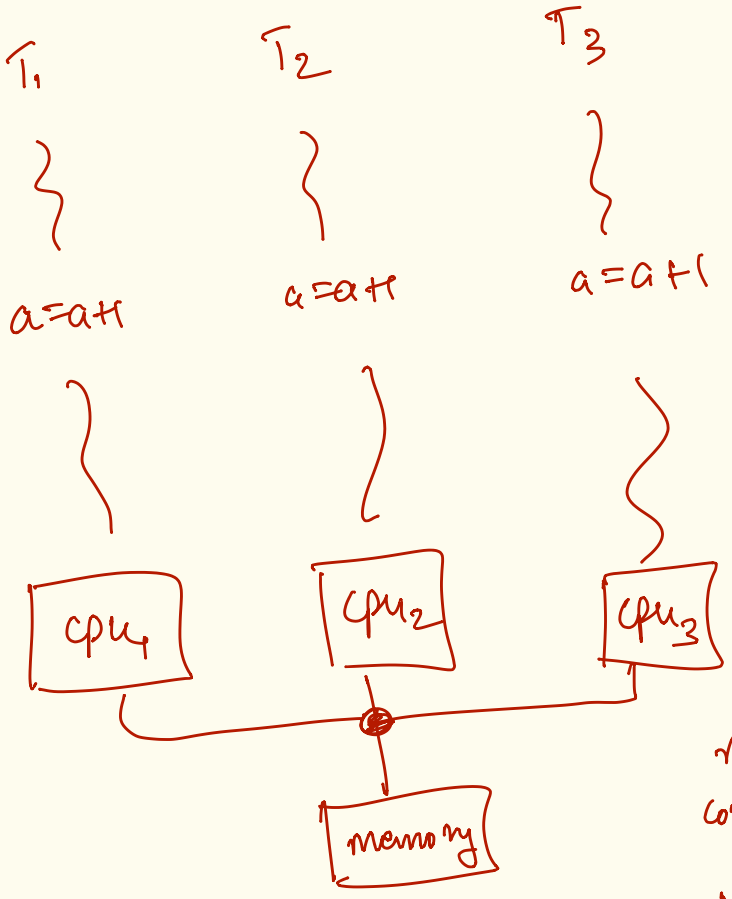
CS744 Lecture 14

concurrency & synchronization



a=23

24, 25, 26

conditions for synchr.

C① concurrency

multiple instances of execution.

C② access to shared memory/objects

race condition

threads should be involved in read — process — write update on shared memory!

```
a=a+1;
mov eax, %addr
inc eax
mov %addr, eax
```

interrupt can cause context switch

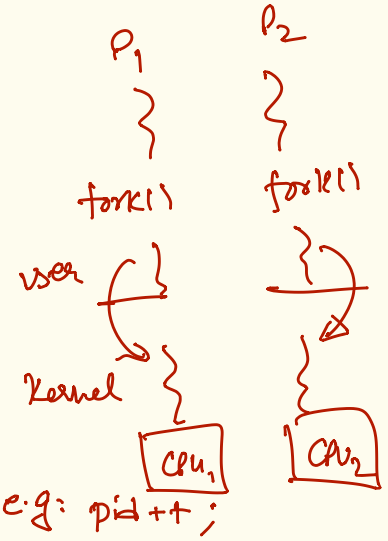
situations of concurrency

① multiple user processes.

② multiple instances of the OS.

③ multi-threaded execution

threads of the same process

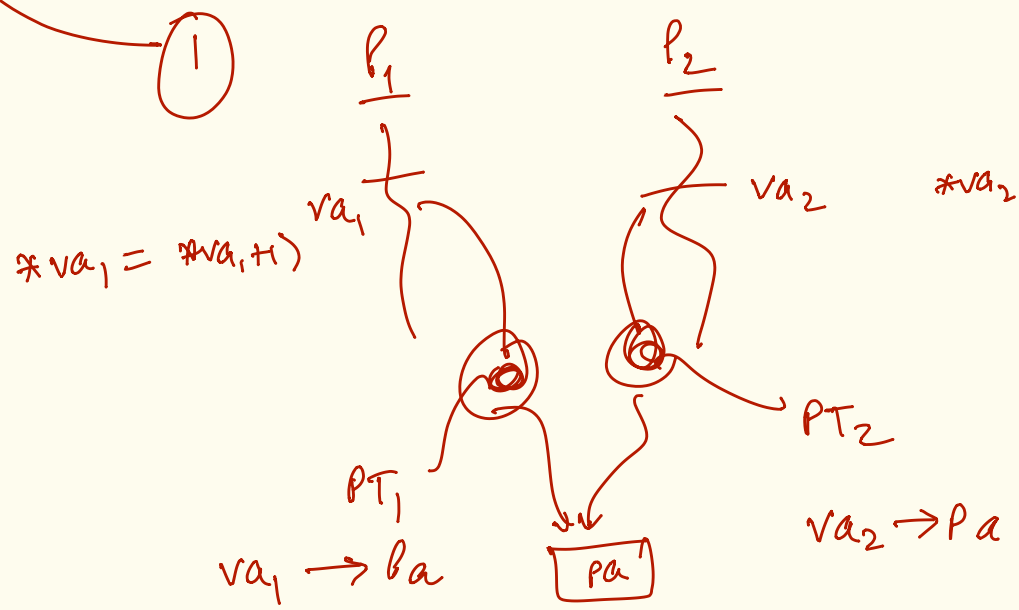


e.g: pid++;

(2) trivially satisfies C_1 & C_2

OS is concurrent & shares all data/memory across instances.

(1)



shm_get
shm_put
shm_add
shm_del

Shared memory related system calls.

(*)

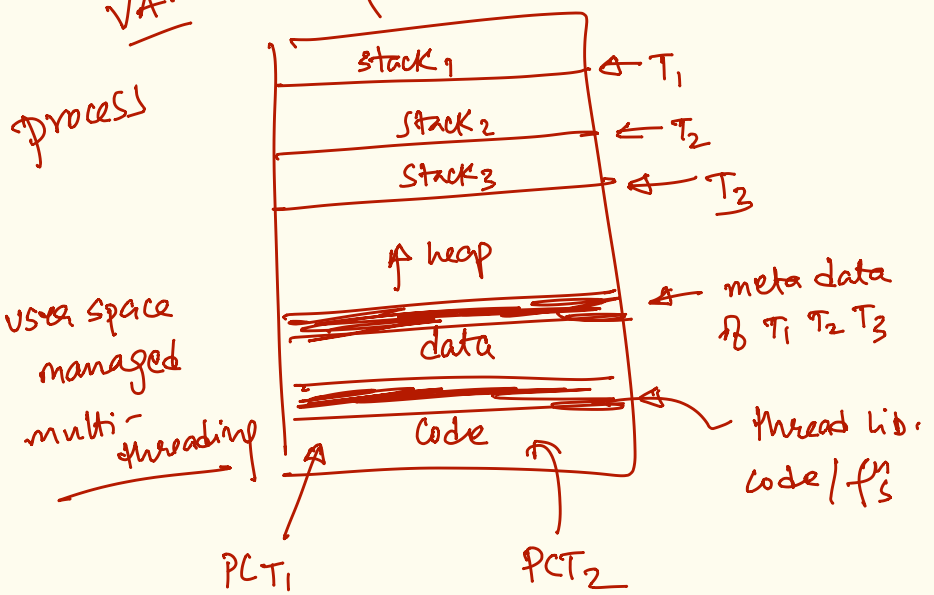
Key = shm_get ();

mmap (Key, size, NULL);

(3) multi-threading!

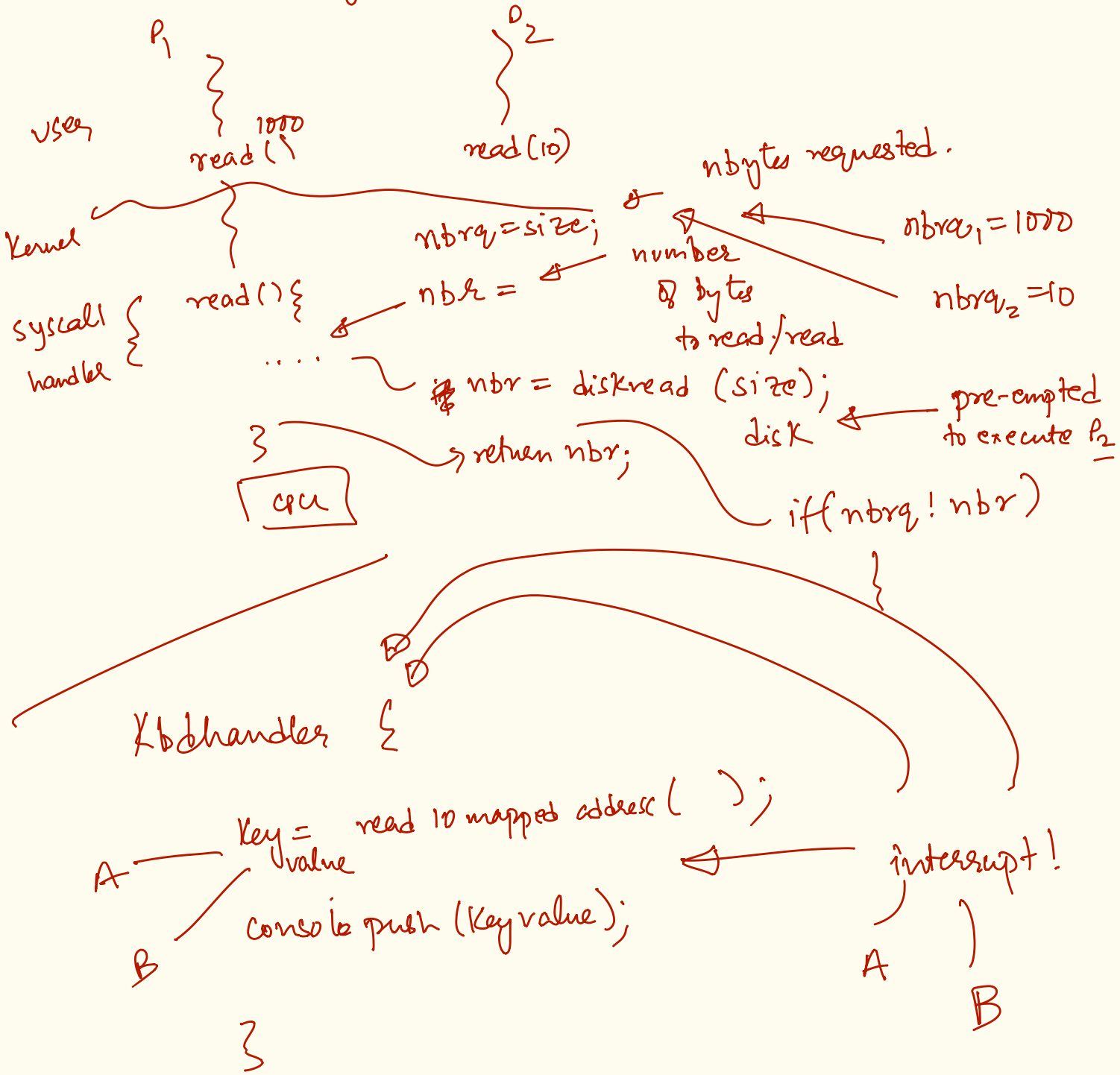
multiple threading using (Kernel-supported threads)

each "process" shares the entire page table except mappings to stack



case (2) multiple instances of OS on single CPU!

PCB: tgid – thread group id
pid – process id

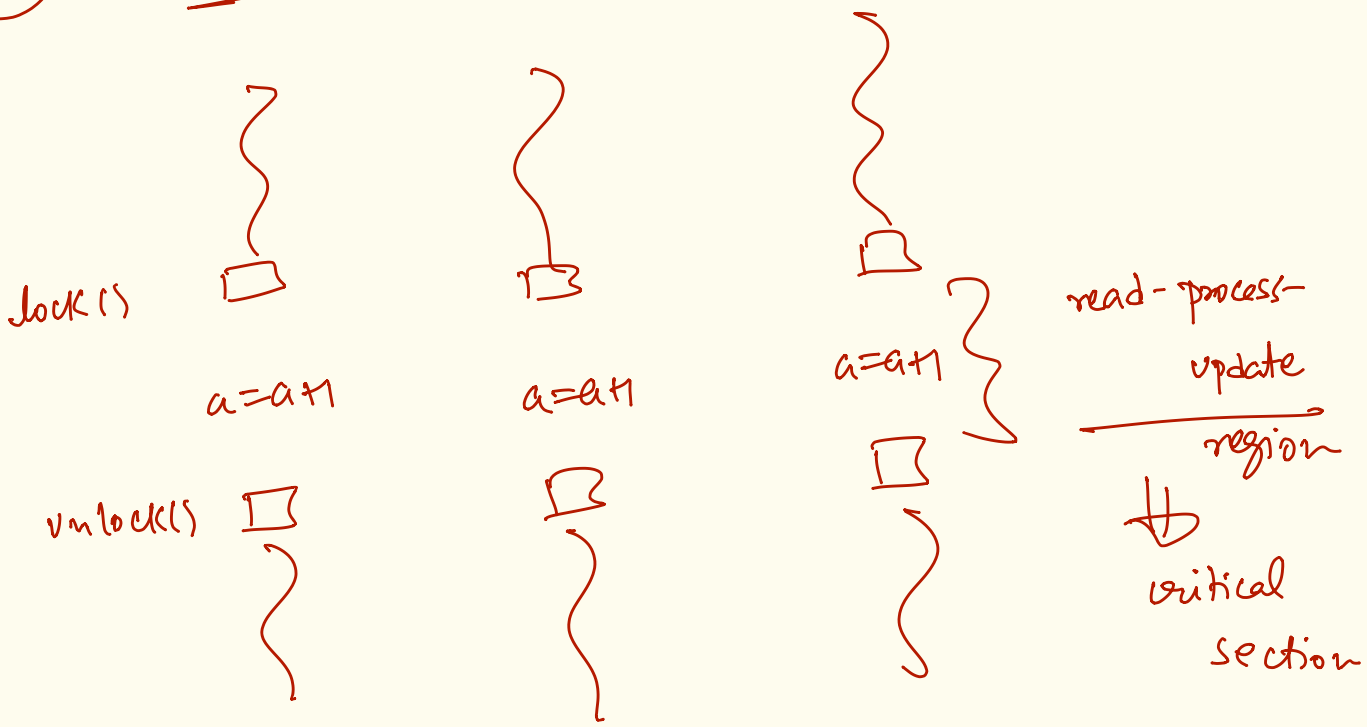


Solutions within the OS!

① disable interrupt

② disable pre-emption.

3 locks



(*) how to design & build locks? (different types of locks)

- spinlocks
- mutex
- condition variables
- semaphores
- barriers.

OS + ISA support.