CS 219 Lecture 23

concurrency & synchronization

① multiple threads / instances of execution ~ concurrency

② shared memory access ~ synchronization.

Lab 8 ~ clone ~ system call

xv6

pid

① tgid ~ thread group id

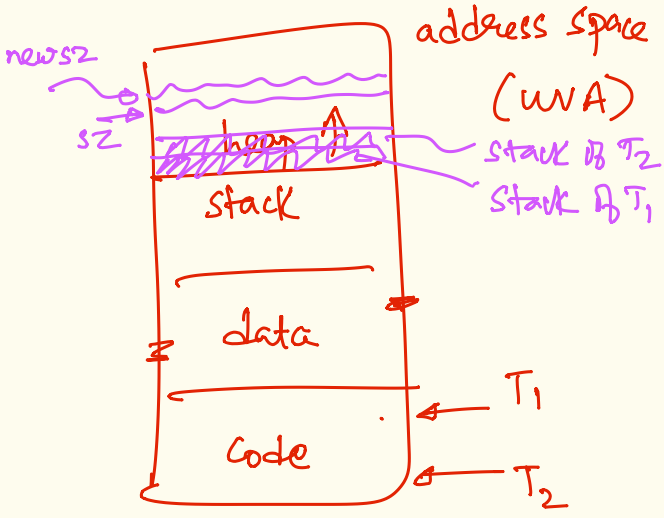
⋮

if pid == tgid

main process.

all process of a thread group have the same tgid.

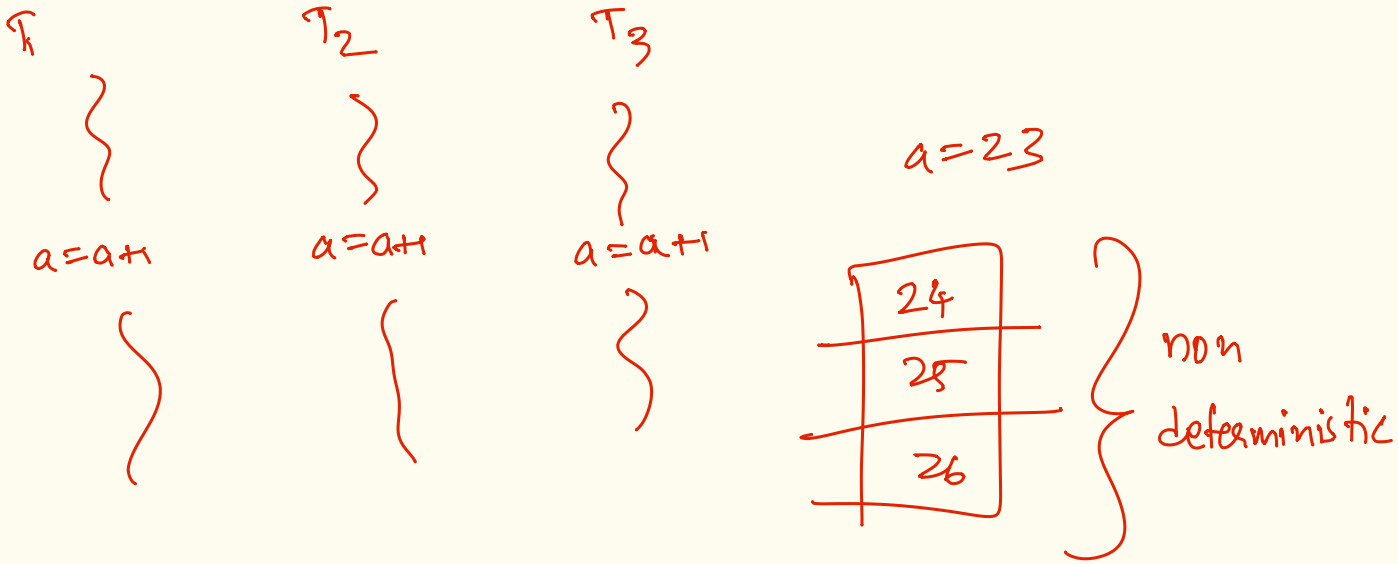
② all processes of the same tgid share the same page table!



pthread ~ C-library

③ system calls need to be tgid-aware!

— malloc / sbrk, exit, ...



(*) what are kernel examples of potential non-determinism?

(i) malloc/sbrk

- find a free page
- sz = sz + size of new alloc

user library or the system call:

~ choose same VA for multiple mappings!

~ choosing the same PA (phy. page) for 2 VA mapping

~ sz is overwritten
↳ wrong size of heap!

(ii) scheduler

~ pick a new process

~ schedule it.

runqueue is shared!
ready

(iii) fork()

~ getfreePCB();

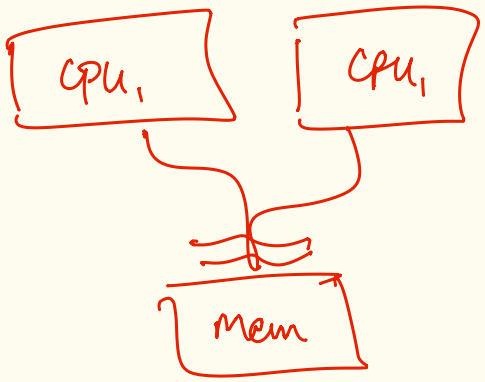
(#) a = a + 1 ⟹

mov ebx, (%addr)
add ebx, 1
mov (%addr), ebx

for the cpu

why 24, 25 & 26 may happen!

most / all memory access + processing (change/ compare) is a multi-step - process



(*) read → modify → write

check

interrupt

interrupt

potential to break atomicity condition

all-or-nothing execution: