Lecture 11

07 August 2018 10:37

Lecture 11. 7th Aug 2018

OSTEP

≡ ① virtualization

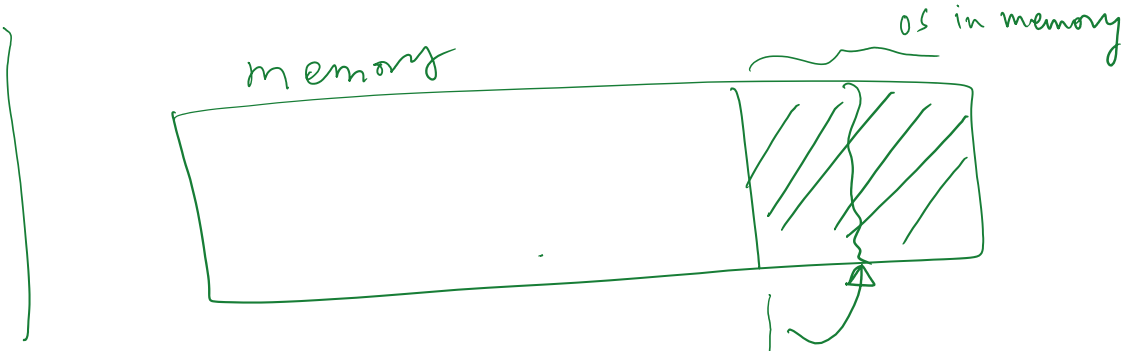
② synchronization

③ persistence

① Kernel threads

- schedulable! (have a PCB)
- to do Kernel (optimization) work
- typical PCB setup (later)

② interrupts — vmlinuz.x.x (a.out)



(i) stop execution of CPU

S/w to privileged mode / Kernel mode — h/w supported

& execute interrupt handler

memory address for fn. schedule

jmp #schedule

(ii) IDT interrupt descriptor table

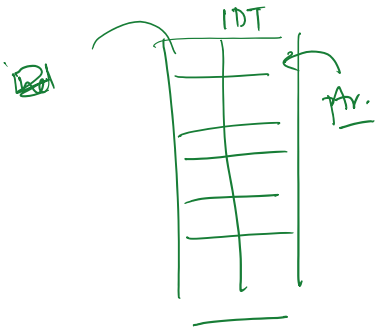


table of fn. ptrs. (memory addresses)

interrupt handler / ISR.

③ system call / ABI

- mechanism to specify system calls arguments

- mechanism for safe entry/exit
- mechanism for identifying which handler
- mechanism to "invoke" / call a system call.

arguments

interrupt
- explicit s/w interrupt
  - int 0x80
- syscall / sysenter / sysexit

in kernel mode
in handler of 0x80 interrupt

(*) x86

mov eax, #int.no
int 0x80

ebx
ecx
edx
:
pass arguments

eax ← holds return value

Wrapper fns. — std 'C' library