Lecture 14

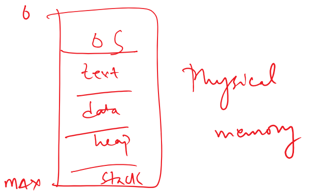
14 August 2018 10:36

Lecture 14. 14th Aug. 2018

(0) memory virtualization recap + design 0.

the address space abstraction (virtual)

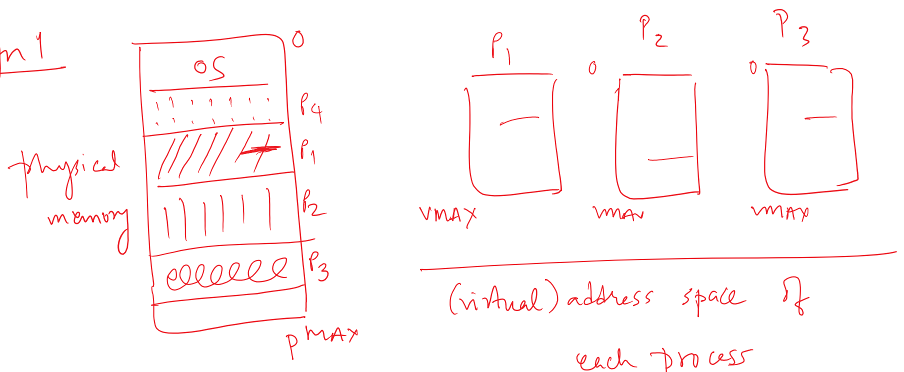
- full addressability
- protection & isolation
- us (0 to max)
- flags (r/w/x)
- efficient!



"one-to-one" mapping between virtual to physical.

-ve: performance is poor. (multi-programming)

(1) design 1



(virtual) address space of each process

assumptions:

- vmax < pmax
- 0 — vmax - all allocated to processes

1a. int x = 23; → $\overset{\text{pre-processor}}{\text{compiler}}$ generated

mov eax, # mem | mov eax, (#mem + offset)

offsets added to programs via compiler / pre-processor

static offset / relocation technique.

- need for recompilation on offset change.
- what is the offset?

1.b. MMU — memory mgmt. unit (part of the CPU)

h/w based translation!

Key: v2p address translation.

– base + bound registers technique

dynamic reallocation technique:

– two regs. base

bound.

– on CPU: every mem. ref.
w/ mmu is "base + virtual address" = PA.

– if PA < (bound + base)

all okay,

else

exception land.

→ H/w support reqd:

– two regs. – base & bound regs.

– (ISA support) execution path of CPU w/ mmu.
integration

ability to
– raise exceptions / interrupt

– mechanism to register handler

– ISA support for privileged mode execution
to update registers.