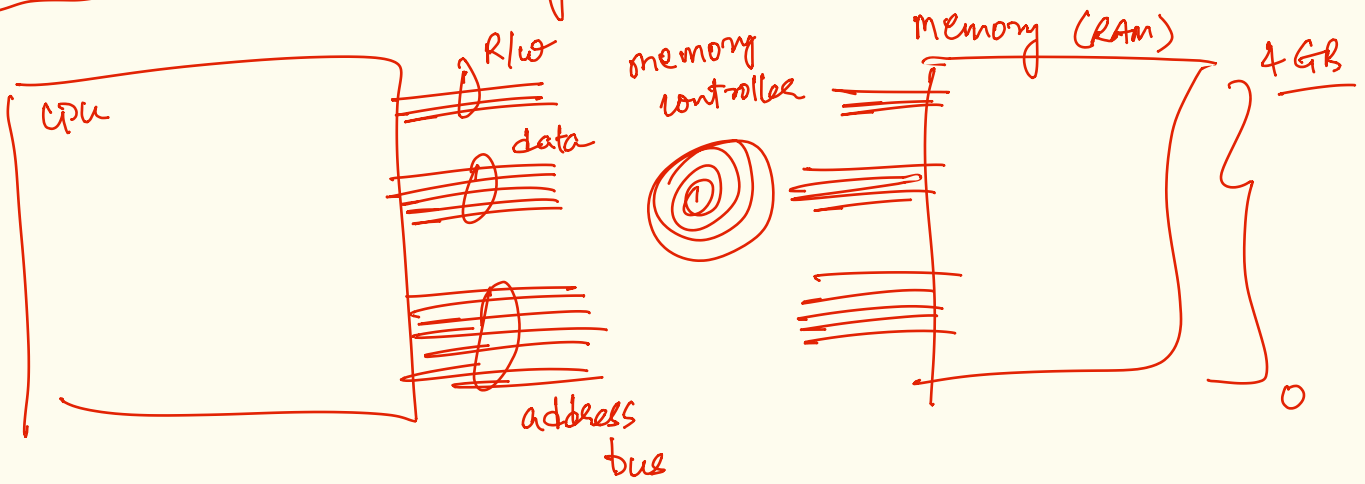


Lecture #10

memory virtualization

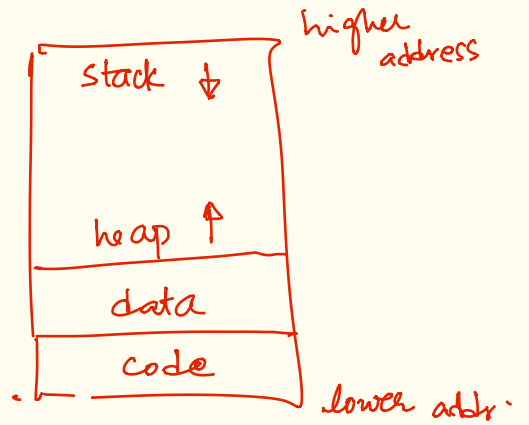
1.9.2023



- sequence of 0s & 1s specifies an address/location to access

(i) processes need memory - why?

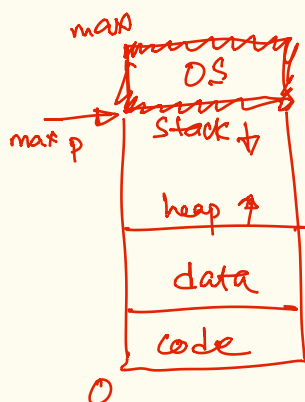
- data (variables)
- code (instructions)
- stack (runtime seq. & args & return values)



(ii) requirements:

- isolation
- zero-starting linearly addressable (full capacity) memory region.
- efficiency
- ! transparency.

design 0:



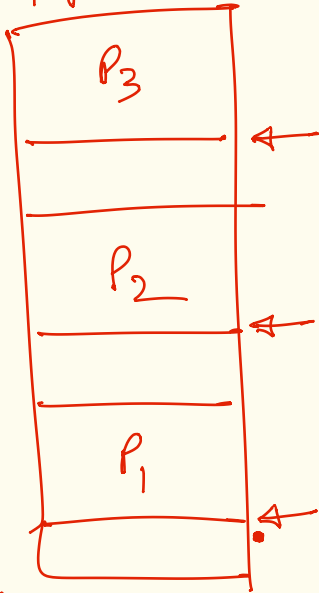
pro: 1:1 mapping of alloc. to reqs.

- con: context extends to all physical memory.
- switch of context will have to write/read all context from disk!

design 1 :

h/w needs to support mapping/translation

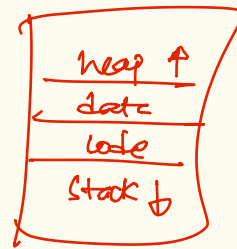
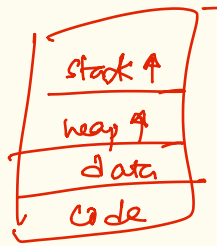
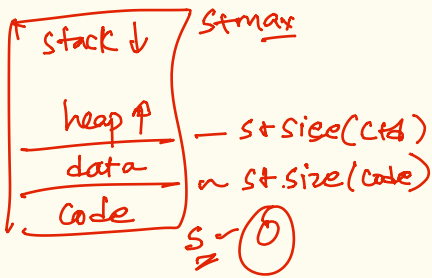
physical memory



+ve : efficiency improves compared to design 0.

- multiple processes can be allocated memory simultaneously.

-ve : assumption of processes not using full addressable range

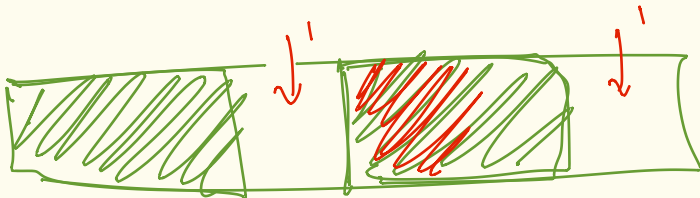
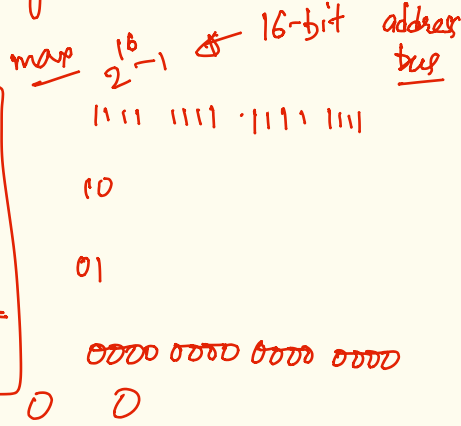
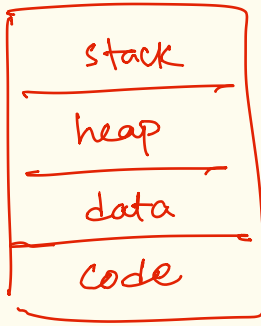


usage efficiency!

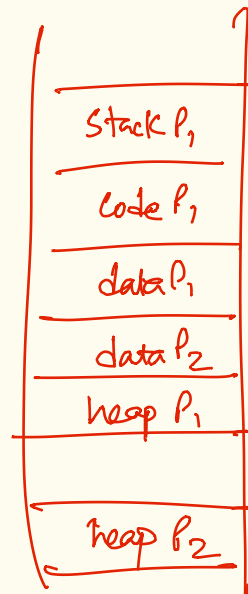
does every process use all allocated memory?

design 2 : segmentation

process view of memory (logical) address space



internal (fragmentation) external



phy. mem