



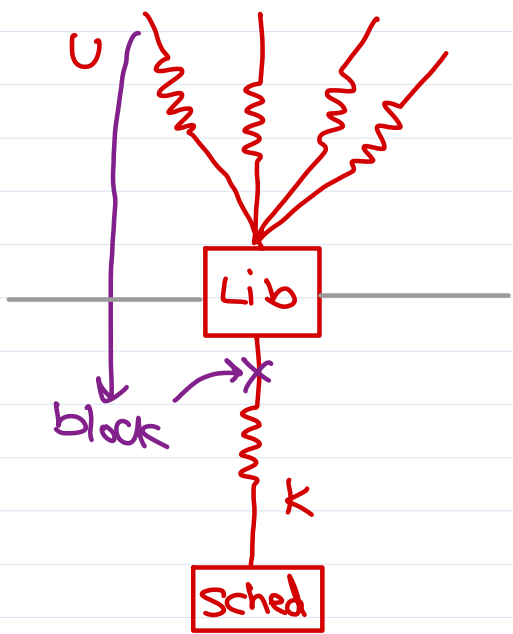
Embedded Operating Systems

Trainer: Nilesh Ghule

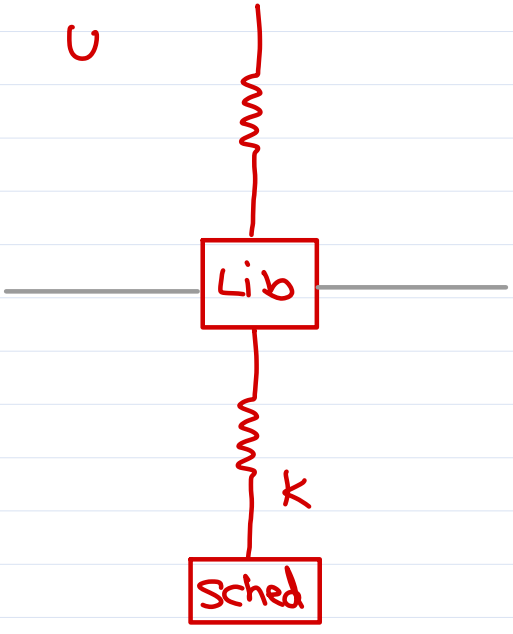


Threading model

m:1



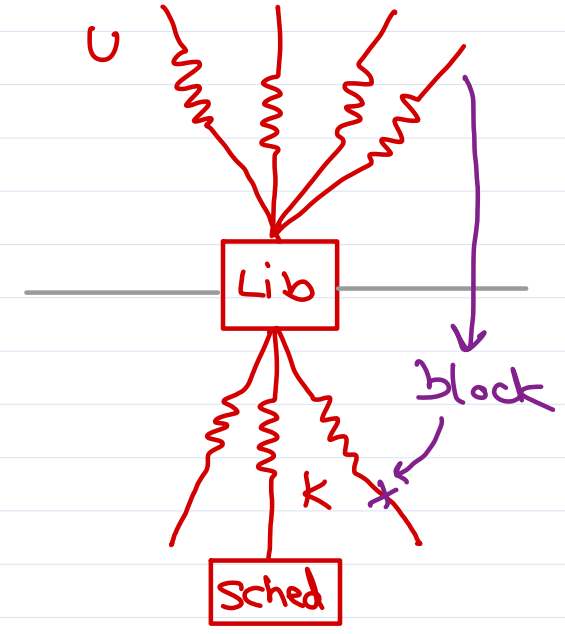
1:1



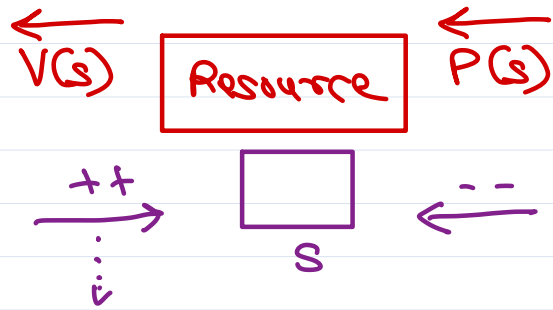
Two-Level model

m:1
+ 1:1

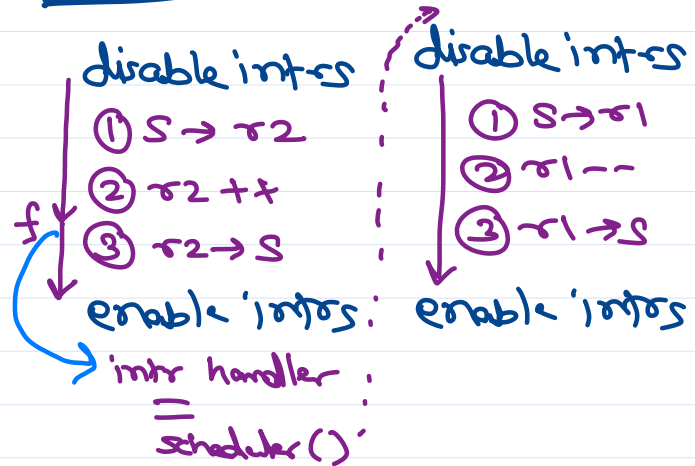
m:m



Spinlock



Solution 1



* Disabling intrs in critical section will avoid preemption of the process (in that section), so that sync obj will not corrupt.

* However disabling intrs will increase intr latency.

* In Smp systems disabling intrs will disable them only for current CPU, but other CPU can still modify sema. So this soln will not work in multi-processor system.

Solution 2 : Spinlock

- hw based sync mechanism
- uses bus holding instruction
- available only in kernel space.
- if spinlock is already locked, current thread keeps running - busy wait - until owner thread unlocks it.

- init: $lock = 0;$

- lock:

$while (lock == 1)$

$lock = 1;$

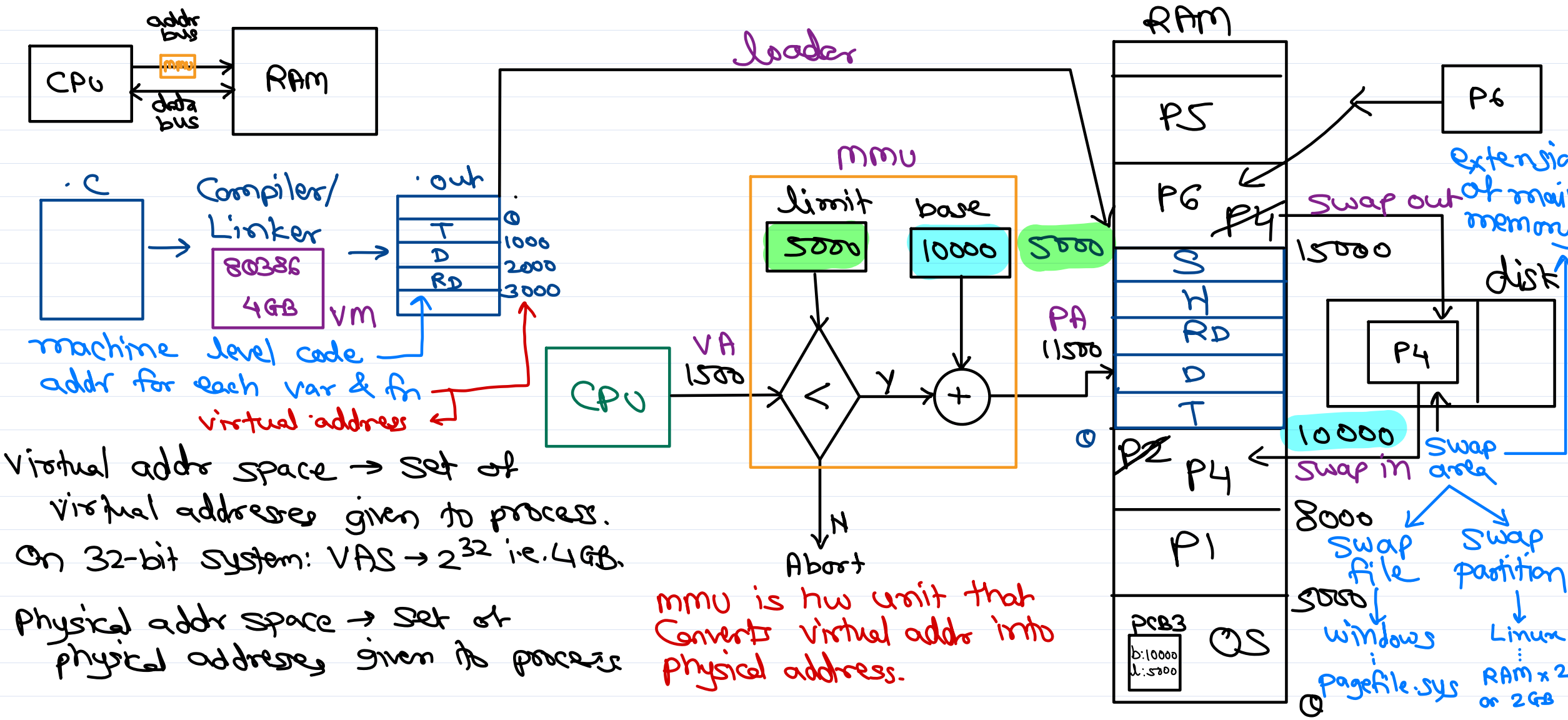
- unlock:

$lock = 0;$



Memory Mgmt

CPU always execute a process in its virtual addr space.



mmu is hw unit that converts virtual address into physical address.





Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>

