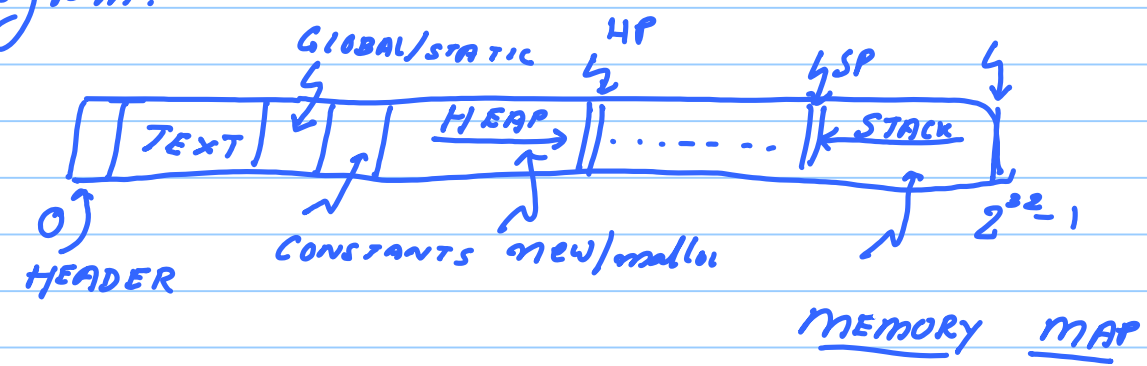


Virtual Memory (26 Oct)

(ELF format)

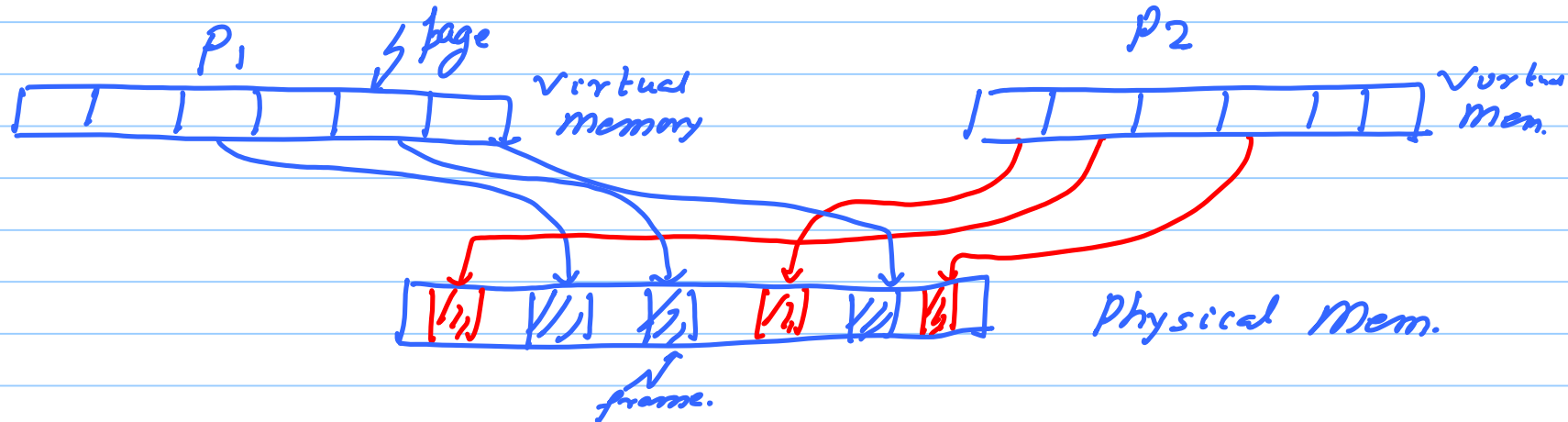
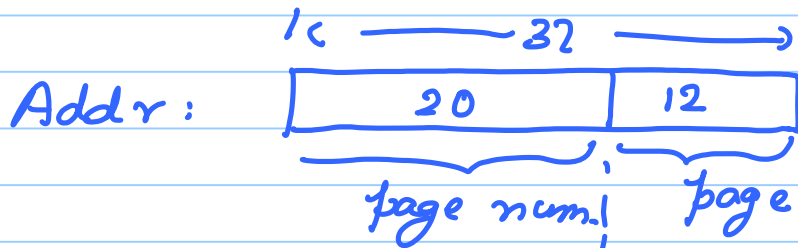
Every Program:



* Isolate programs. (No collisions between programs)

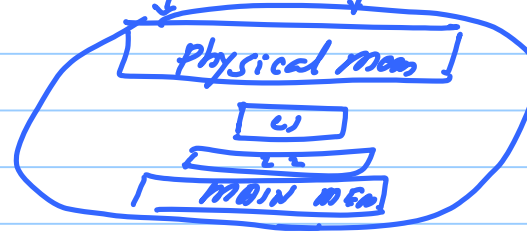
1) Divide memory into chunks of 4 KB each.
(page)

$$4 \text{ KB} \rightarrow 4096 \text{ B} \rightarrow 2^{12} \text{ B}$$



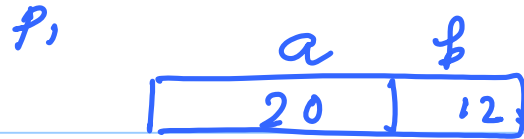


It is possible:
 $A = B$
 $(A' \neq B')$

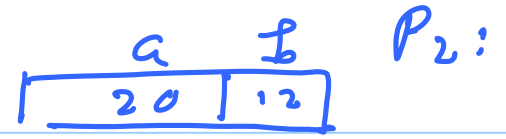


- 1) Every program sees a contiguous view of memory.
- 2) It does not assume the existence of other programs.

The process of address translation ensures isolation across programs.



$A = ab$



$B = ab$

TRANSLATION

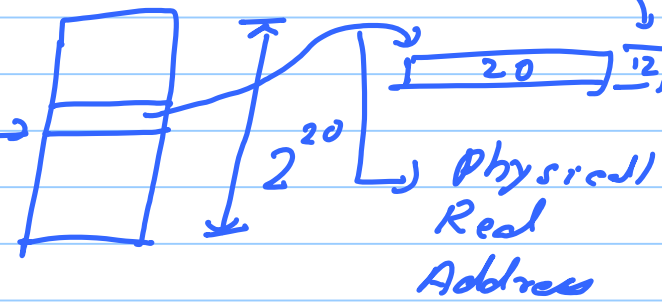
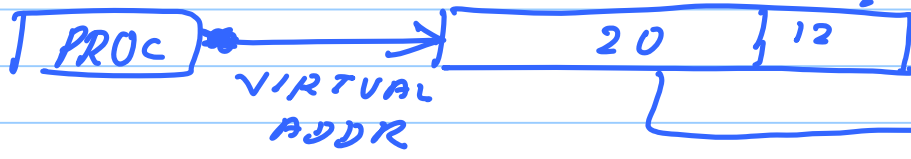
$A' = a'b$

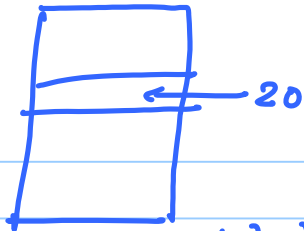
$B = a''b$

$(a' \neq a'')$



TRANSLATION

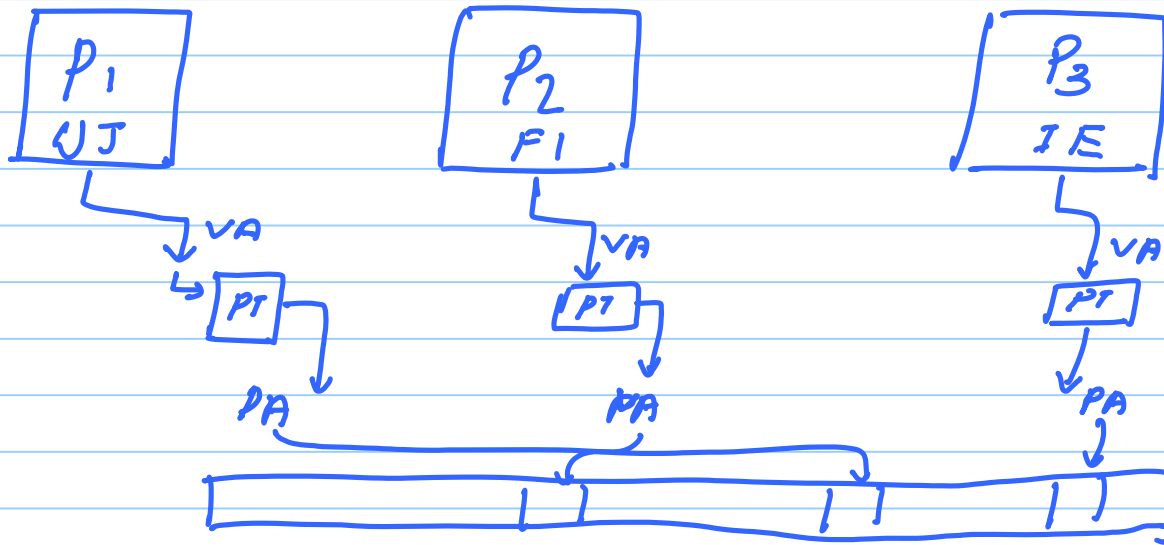




(Page Table)
(PT)

→ $(2^{20} \times 20)$ bits
 ≈ 1 million (2.5) B
 ≈ 2.5 MB

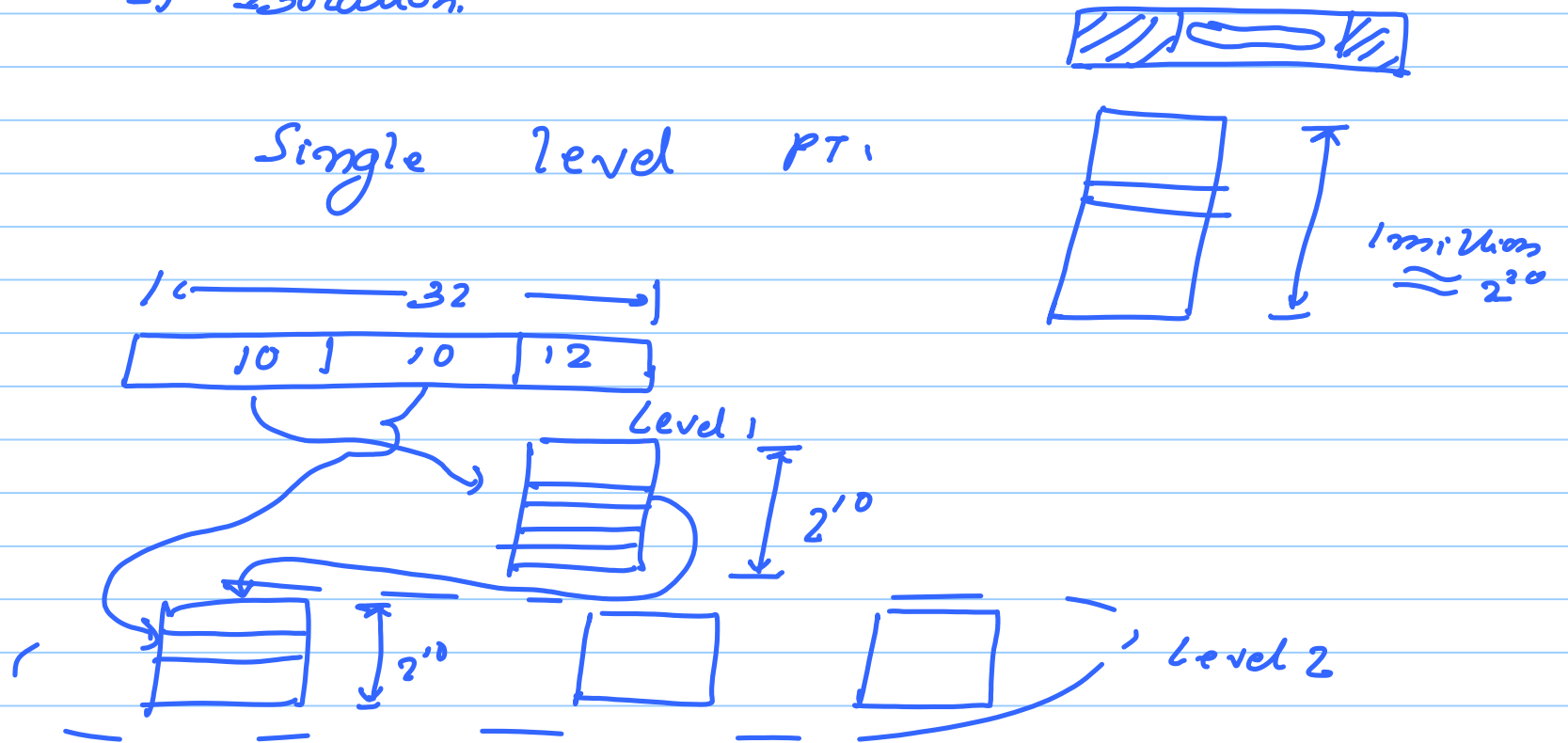
Inefficient: Single level PT



Page Table:

1) Contiguous Virtual Memory
(Non-contiguous) Physical Mem. allocation

2) Isolation.



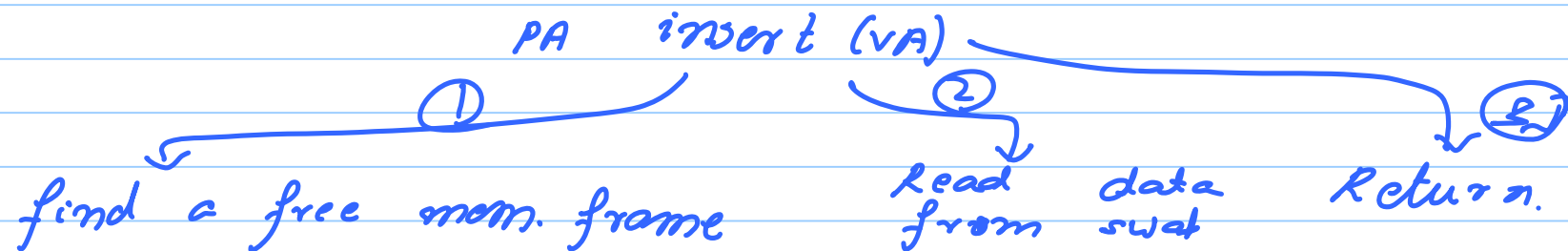
Storage: $\underbrace{2^{10} \times 10}_{\text{Level 1}} + \underbrace{20 \times K}_{\text{Level 2}}$ (Kilo Bytes)

$K \rightarrow$ the number of secondary page tables.

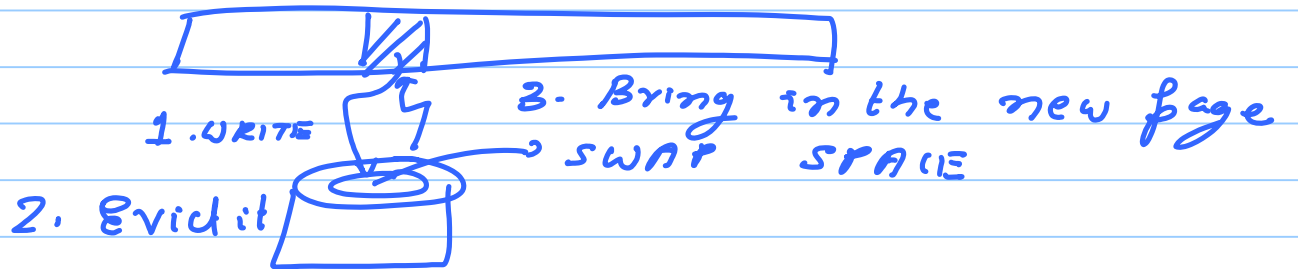
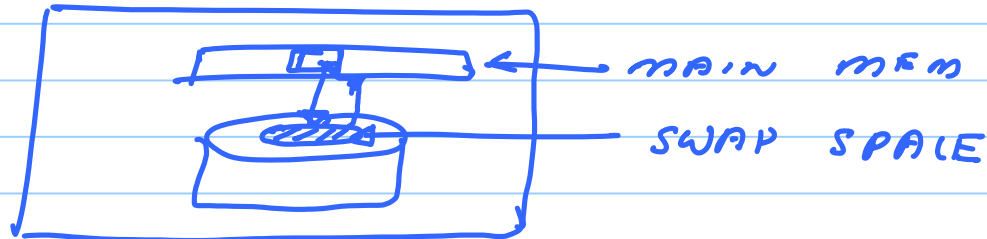
MMU \rightarrow Intel Processors
(Hardware)

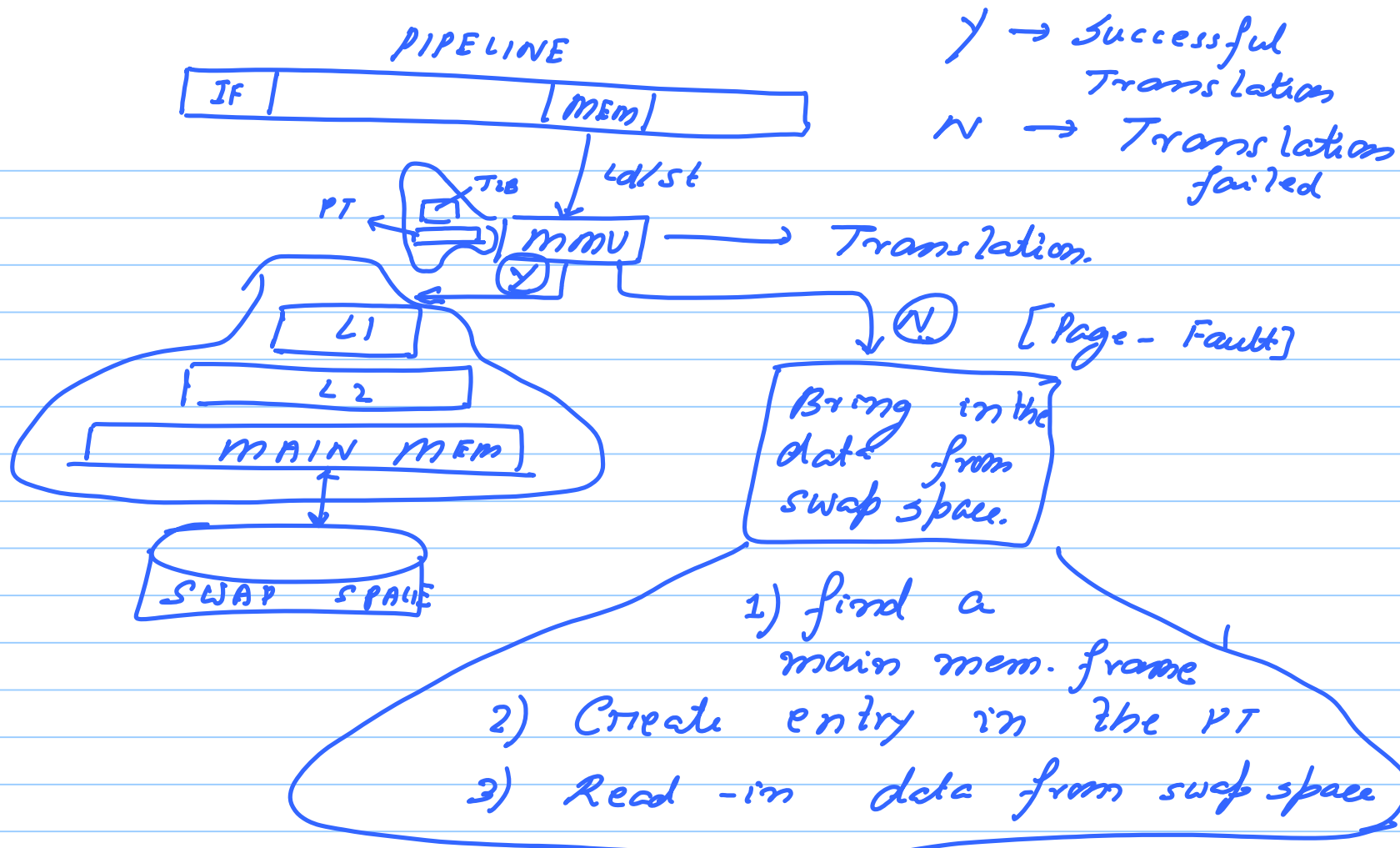
IBM/ARM \rightarrow Software (MMU)

Page Table: PA lookup (VA)



→ cannot find a free frame: ^{space.}
evict one





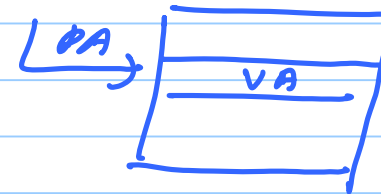
mmu needs to be very fast

Page Tables are slow



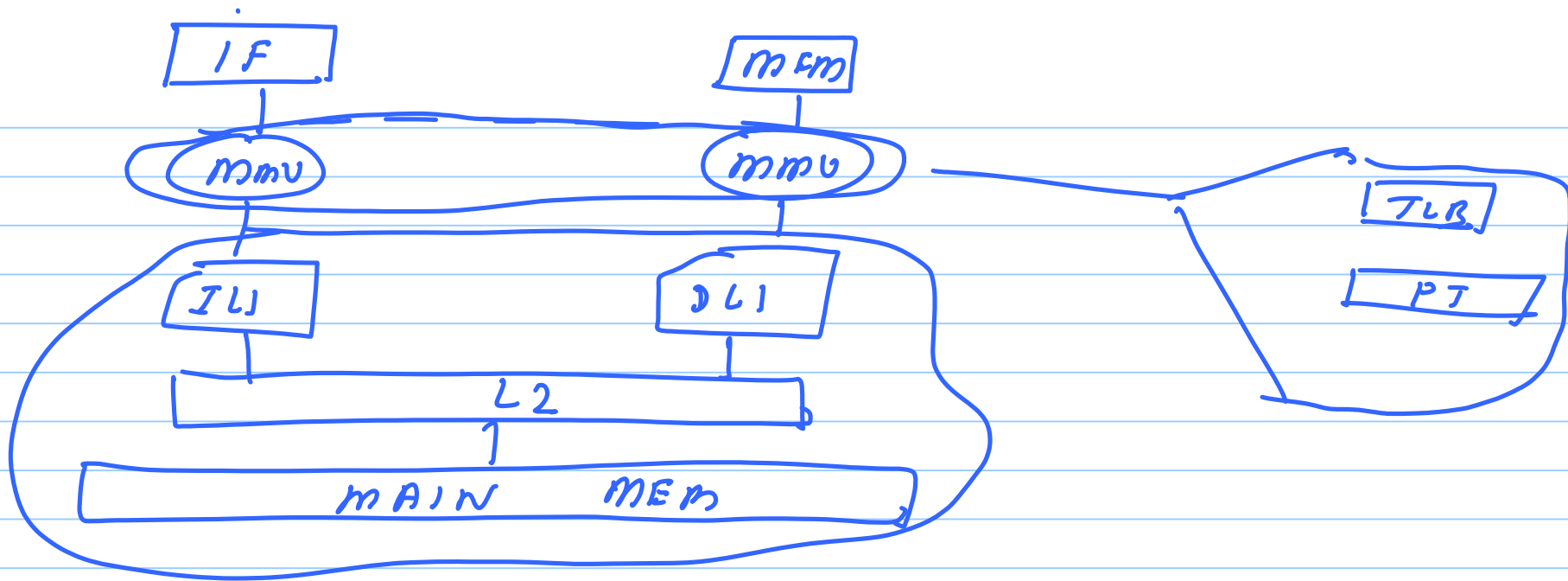
TLB → small cache
for page
translations

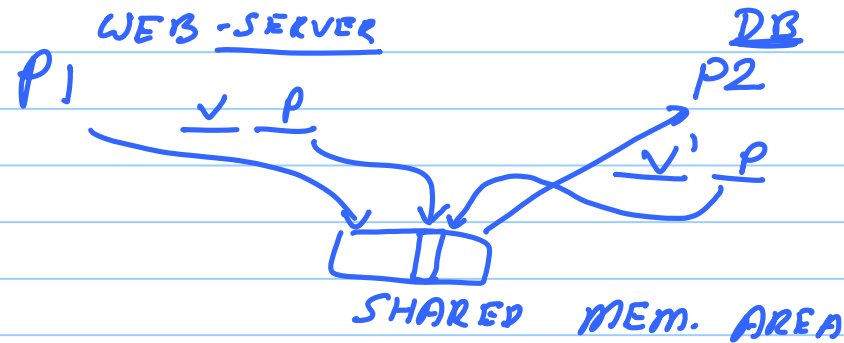
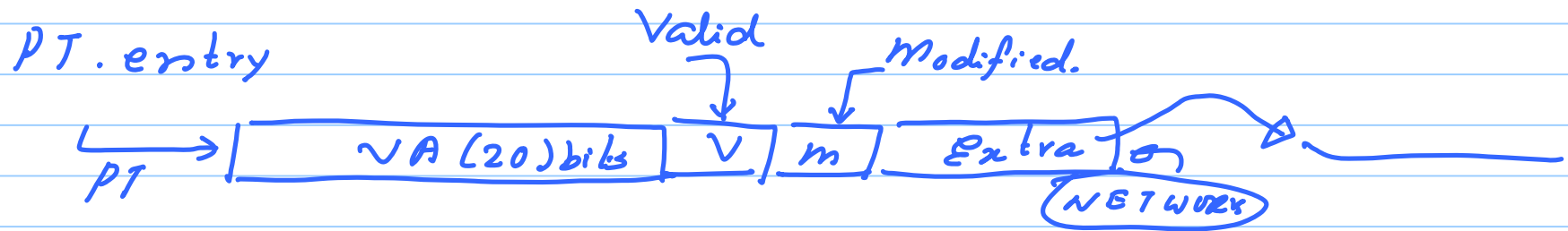
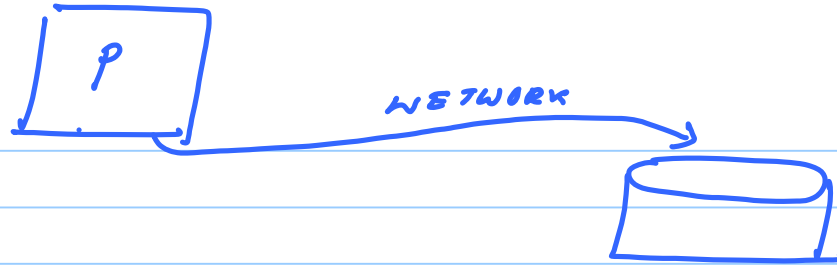
TLB Miss: 30 cycles.
TLB Hit: < 1 cyc



Fully Assn.
64 entry
struc.

(very fast)
(99.9% hits in TLB)





TODO: [(shmmem) call on Linux]
 (shmget)