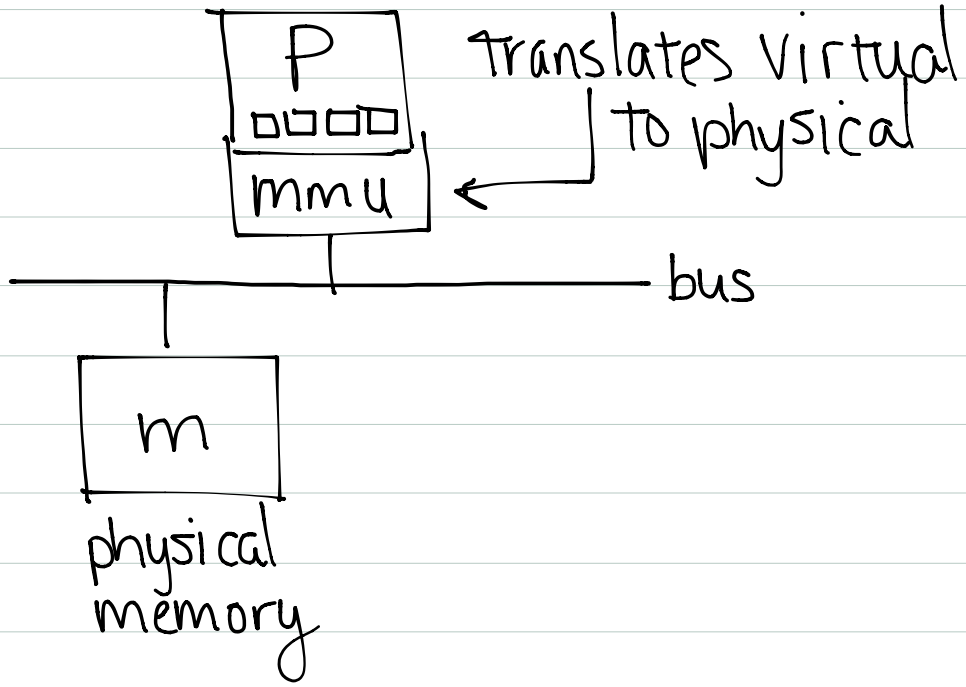
If physical address length is m bits then max amount of phys memory is 2^m (bytes)

Let virtual address size be v bits

max amount of virtual memory is 2^v (bytes)

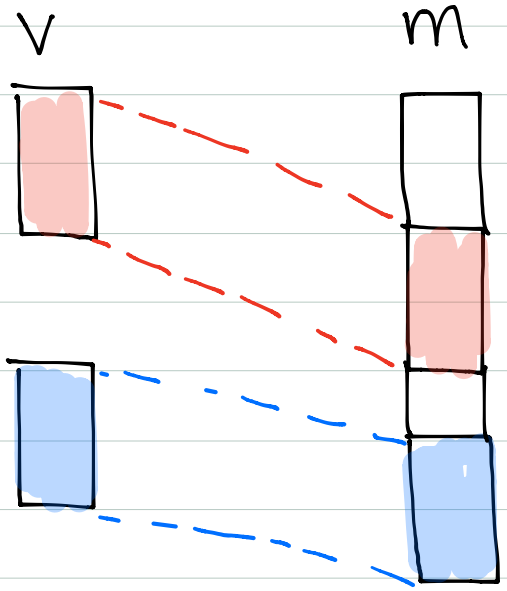
Virtual Addresses
- alleviate problems when compiling and linking
- code doesn't have to worry about where in memory it is sitting when it actually runs.



- MIPS ~ 1-2 virtual to physical translation per instruction

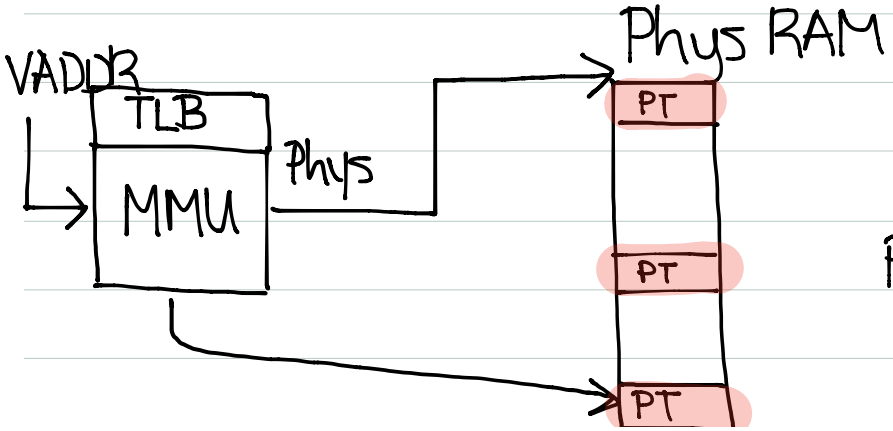
↳ that's why hardware is required

Dynamic Relocation:



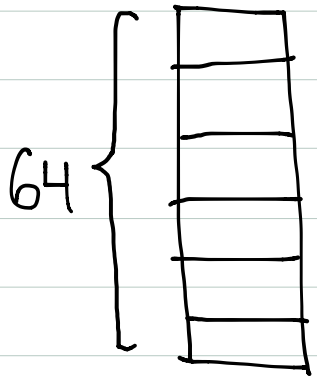
Relocation Register
- holds base physical address of currently running thread

MMU may also have a limit register to check for virtual addresses that are too large.
- throws an exception - segmentation fault.

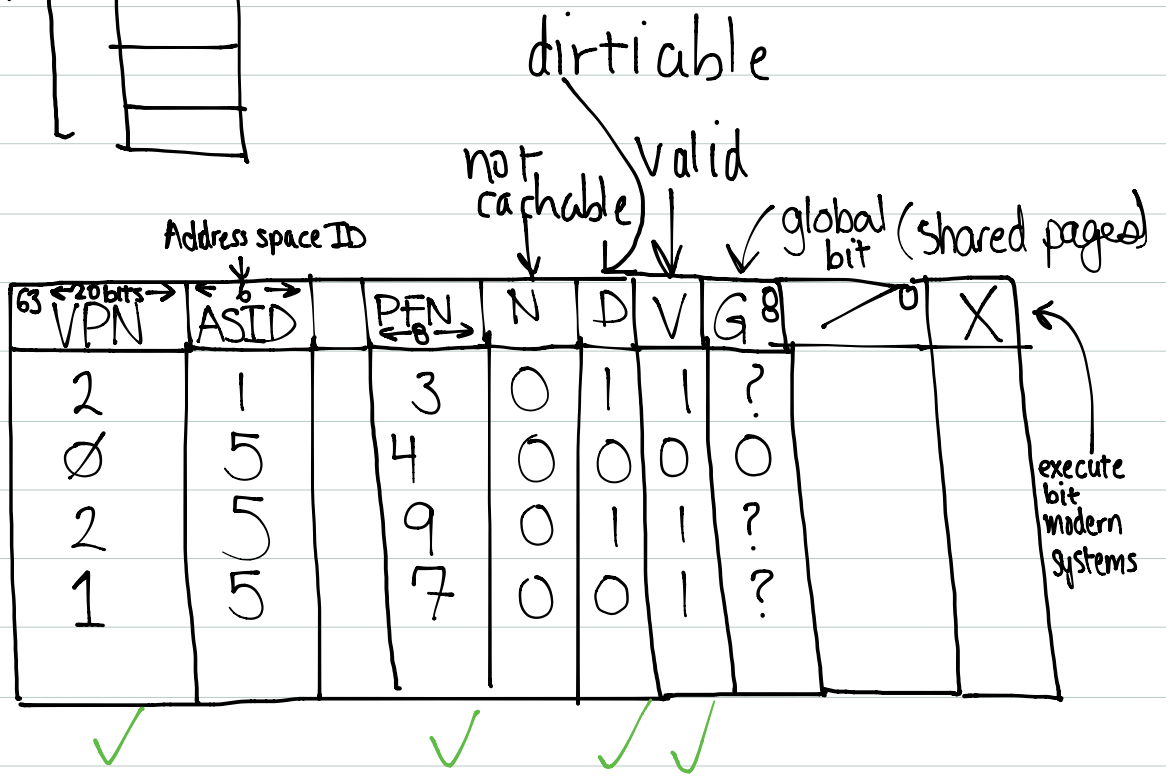


PT=page table

Fully associative TLB



- big cache no good if TLB doesn't cover it.
-



Need to understand!

Address translation using TLBs

VADDR: 0x00009ACD

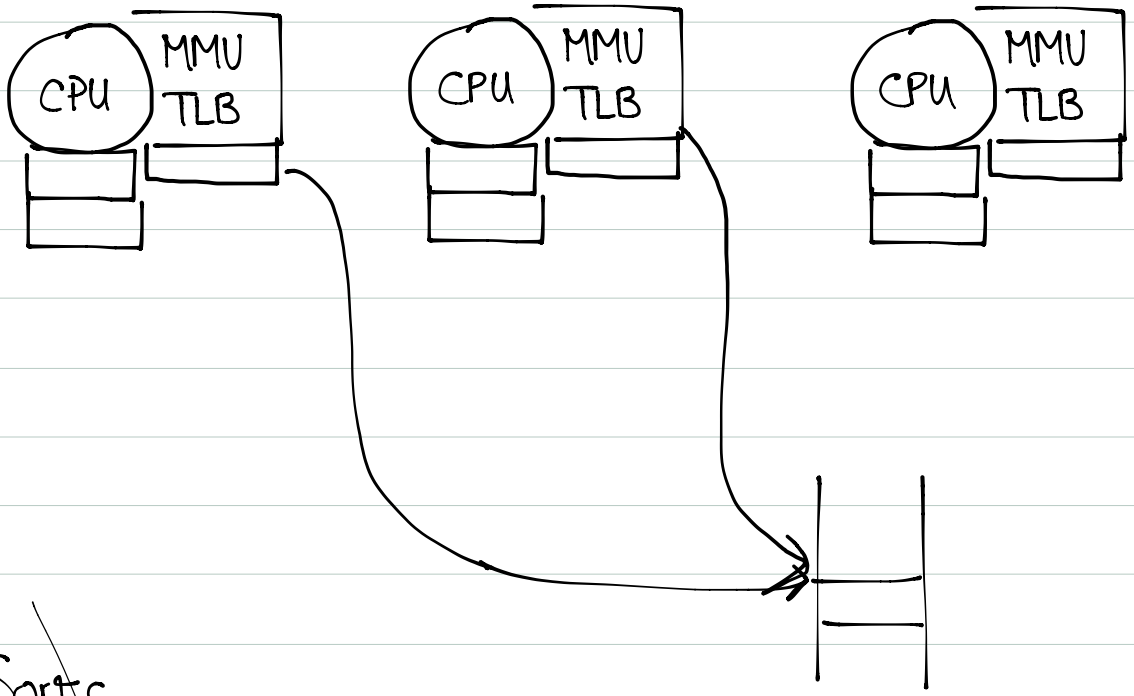
Page size: 4096 - decimal
or 0x1000

$\frac{\text{VADDR}}{\text{Page size}} = \text{VPN}$

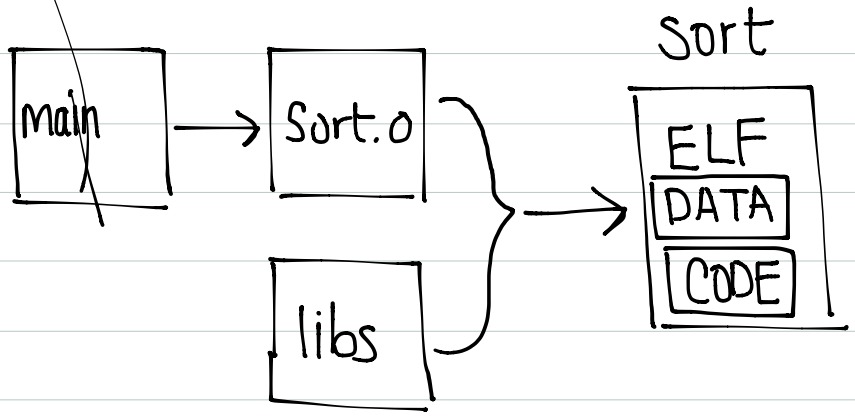
Remainder — ASID

Ex2: 0x00002 FEB
0x00003 FEB

So far, covered TLB for uniprocessor



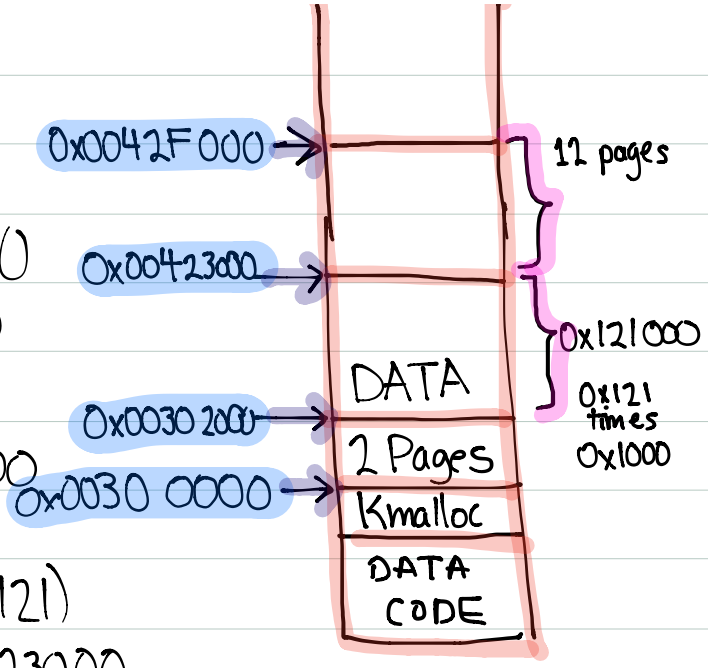
Sort.c



Segmentation

Dumbvm

vbase 1 0x0040 0000
pbase 1 = 0x0030 0000
mpages 1 = 2 (0x2)
vbase2 = 0x 1000 000
pbase2 = 0x0030 2000
mpages2 = 289 (0x121)
stackpbase = 0x00423000



Valid App stack range:
7FFF0000 to 7FFF4000

Translating virtual to physical

v = 0x00401 6A0
since OS161 Page size 4096 (0x1000)

Page 1 (starts at zero)

Page frame # = 0x0031 6A0

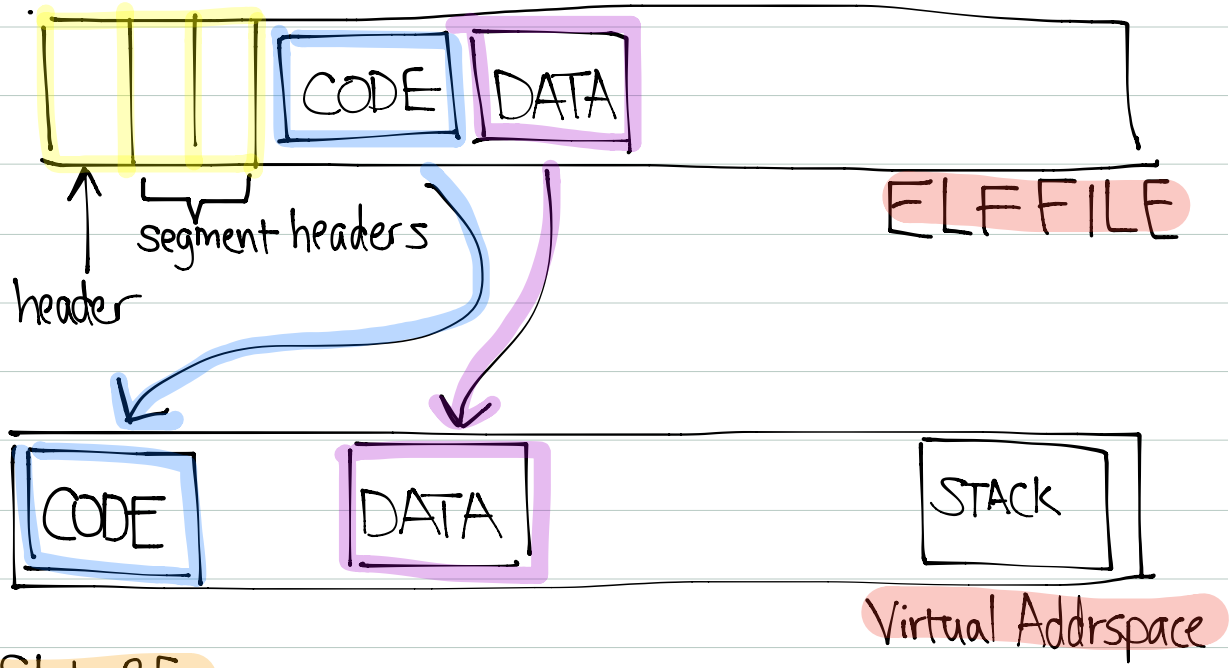
virtual addr: 0x10099220

0x00302000
Page 99
0x0039B220

NOT DONE

0x7FFF329F

ELF Files



Slide 35

.text: the actual instructions
.rodata: z, "Hello World\n"
.data: x, str
.bss: array
.sbss: t1, t2, t3

the ptr of str mutable but the initial value immutable

Useful elf fcns

read_elf
objdump

Shared Virtual Memory

- kernel sets it up, but the two processes may not know they overlap
  - ↳ one copy of vi program (read-only)
- ↳ processes can't tell the difference
- ↳ read-only

- what if we want it mutable
  - ↳ overlapping part in address space
    - not transparent
    - processes ask kernel for their address spaces to overlap

Shared Pages

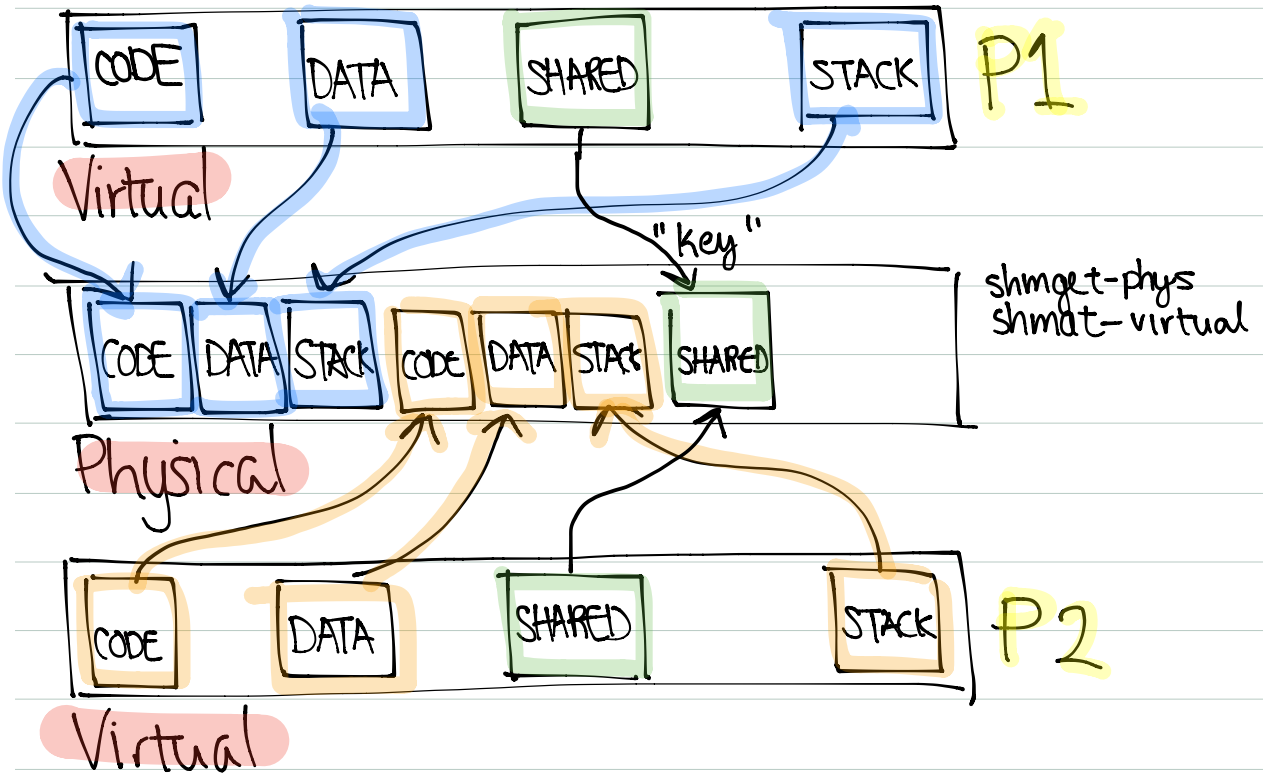
- maintained by page table
  - ↳ entries with same frame number

Segmentation and Paging

- segment sharing
  - ↳ share common segment table

- page sharing

Interface for setting up shared memory



shmget - kernel check if memory (block) with key already made
otherwise create it
returns shmid

shmat - maps the virtual memory to the physical

How do the programs know the "key"?

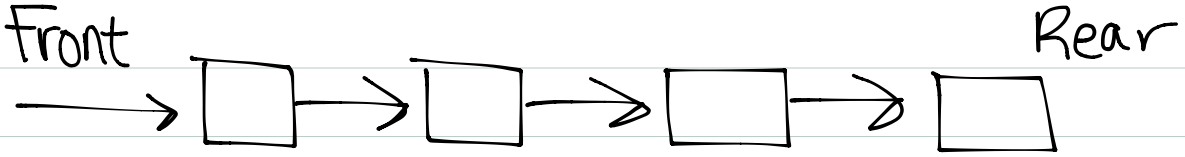
- need to avoid collision!

When we call malloc
- brk - get more memory
- sbrk - shrink memory

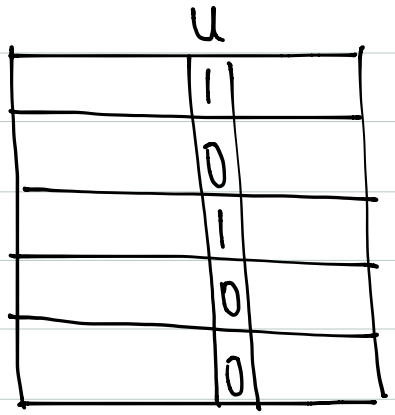
swapping - move data to and from disk
- want to avoid!

90/10 or 80/20 rule
- % time / % data

LRU - Least recently used page replacement



LRU - Is NOT optimal, but is generally better than FIFO



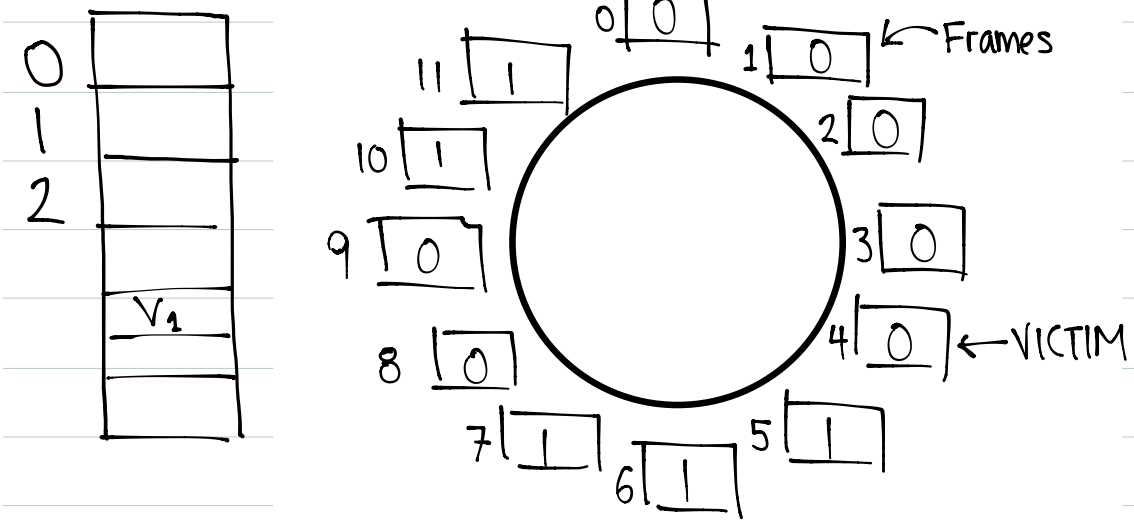
Used bit - 1 for recently used found in TLB entry

When we switch processes, we check used and copy it to the page table and invalidate the TLB.

MIPS doesn't have a used bit in TLB
↳ How do we simulate it?
- Trap then set page bit Page 60 of VM Slides

Valid vs used:
- touched vs good translation

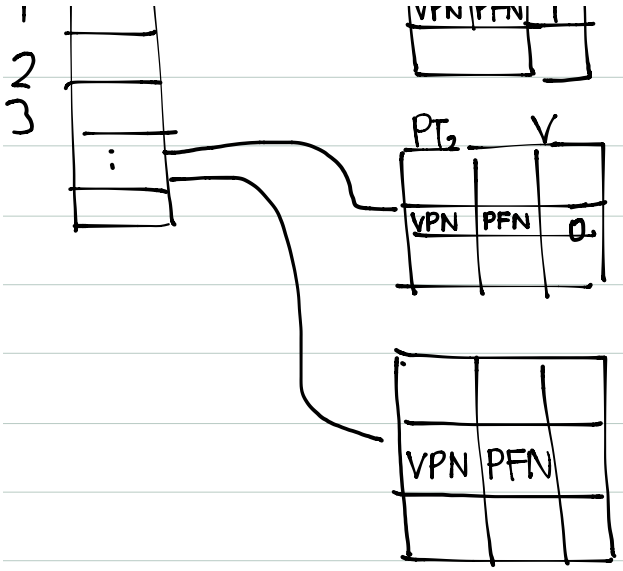
COREMAP
-core is another name for RAM or memory



Clock replacement algorithm

If V_1 is dirty, copy it to disk.

Best thing to store on coremap:


1
2
3
VPN PFN
PT2
V
VPN
PFN
0
VPN
PFN