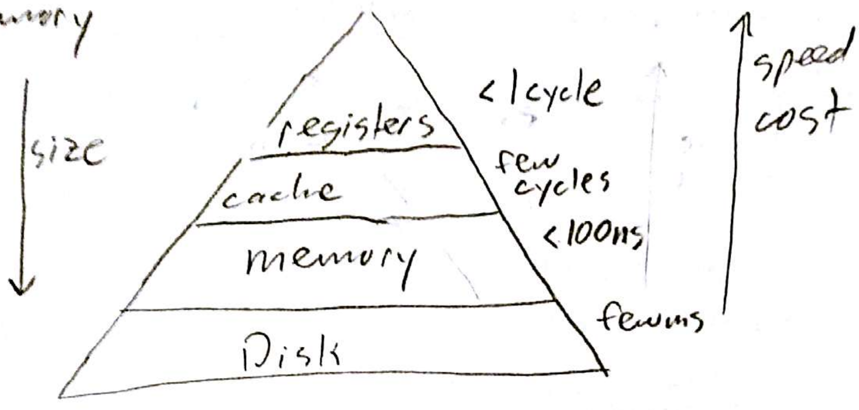
other methods as well
Hashed (for OSs > 32 bits)
Inverted
combined Paging + Segmentation

Virtual Memory



on 3.25 GHz machine
reg = 3.1×10^{-10}
cache ≈ 15×10^{-10}
mem = 1×10^{-7}
disk = 1×10^{-3}

$\boxed{\text{reg is 3000 times faster than mem}}$

Motivation

Previously:
All of a process must be in mem
large process → out of mem :(

But remember:
locality of ref: you tend to access mem around where you last accessed mem.

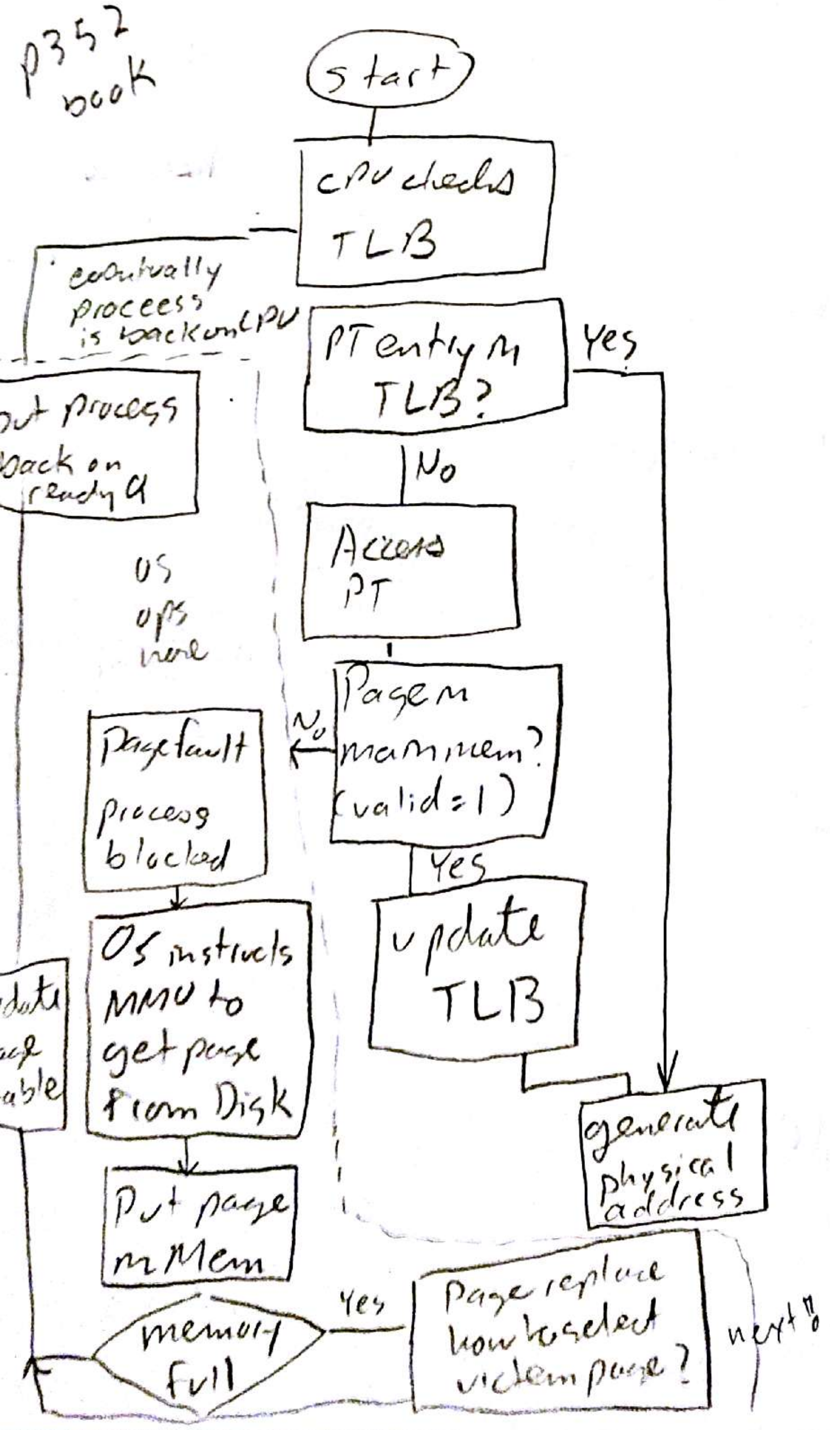
So...
Only need a small part of a process in mem at a time (part thats running)

Idea

Process runs when not all pages in memory

- keep close by pages in mem, those currently referenced (PC pointing to, function just called...)
- keep unreferenced stuff on HD
- Pull in from Disk when needed (Xparent to the user)

p352 book



start
CPU checks TLB
PT entry in TLB?
Yes
No
Access PT
Page in main mem? (valid=1)
No
Yes
update TLB
generate physical address
Page fault process blocked
OS instructs MMU to get page from Disk
Put page in Mem
memory full
Yes
Page replace how to select victim page?
next!!
update page table
put process back on ready Q
eventually process is back on CPU
OS ops here

Disk is 10,000 → 100,000 times slower than mem. Big cost so load big pages when you have to go to disk (4K → 1Meg) ictual Mem operations.

Detect page fault

choose free physical page

OS algo. if none free OS chooses victim page(from current process)

Bring page to mem from disk

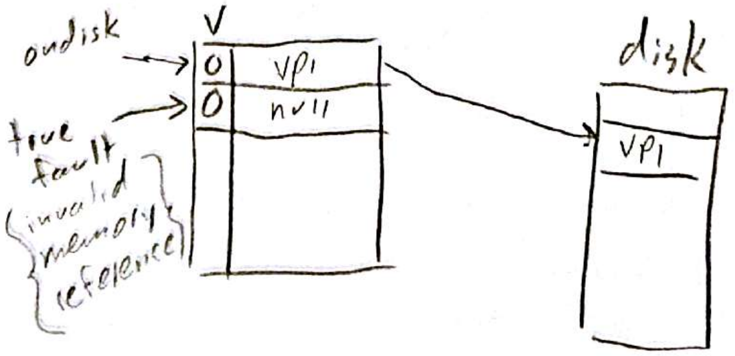
Need H/W + Software

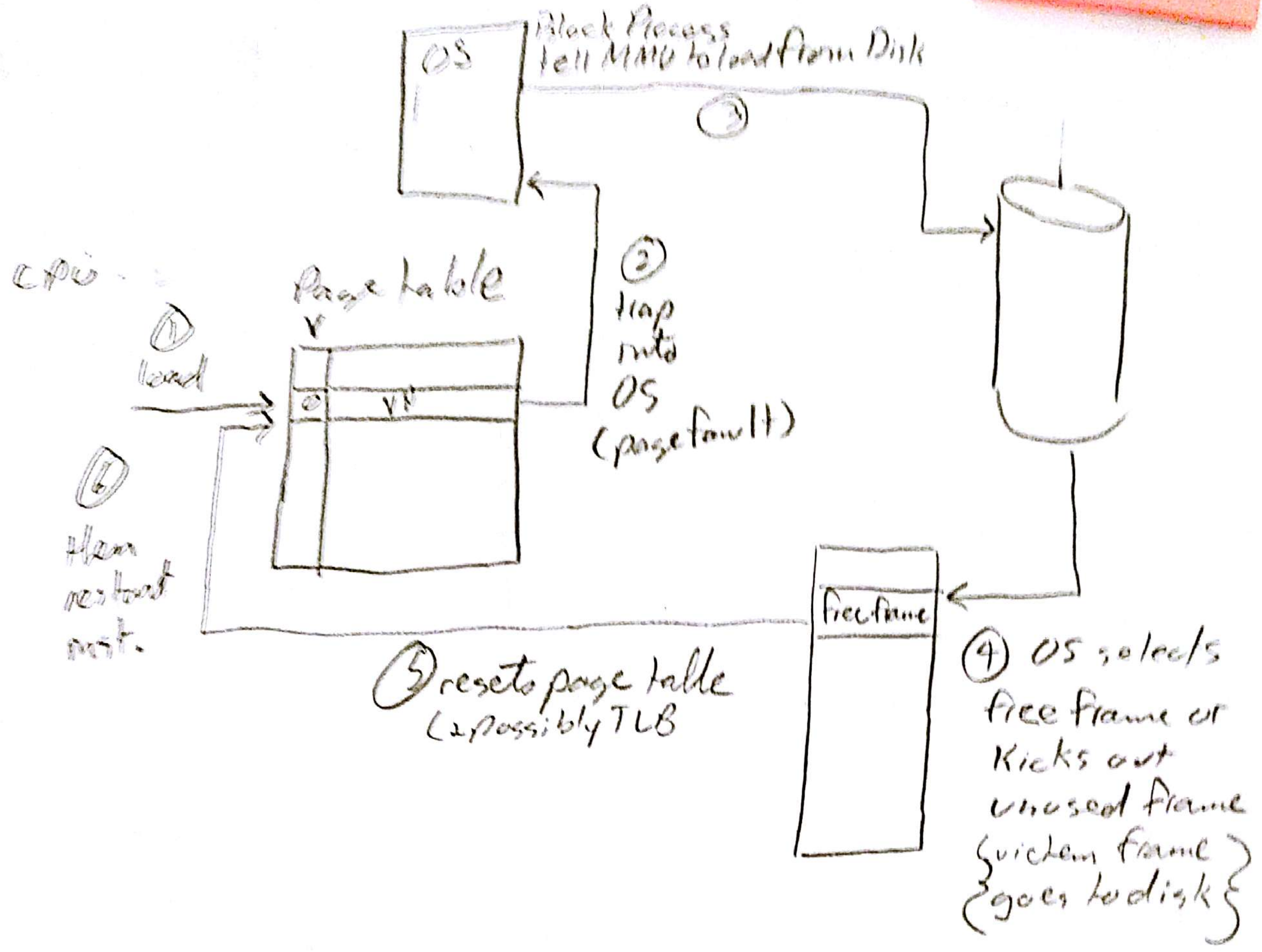
→ valid bit - if 0 then pagefault.

V=0 if page on disk, store disk location in page table

V=0 page fault, referencing results in a trap to the OS

In OS page fault handler, check to see if fault is caused by ref to true invalid page or page on disk




Block Process
tell MMU to load from Disk
3
OS
CPU
Page table
v
1
load
0
vr
2
trap into OS (pagefault)
6
then restart inst.
free frame
5 resets page table (+possibly TLB
4 OS selects free frame or kicks out unused frame
{victem frame goes to disk}

when to bring pages from disk?

on Demand? Start up with no pages loaded
Wait until a page must be in memory

Request - user speifies, user manages memory
by hand :C, User not expert, user
can use up all memory

Prepaging - load page before needed, JIT,
very effecient, when one page refed
bring in next, or predict what is
needed (hard to do.