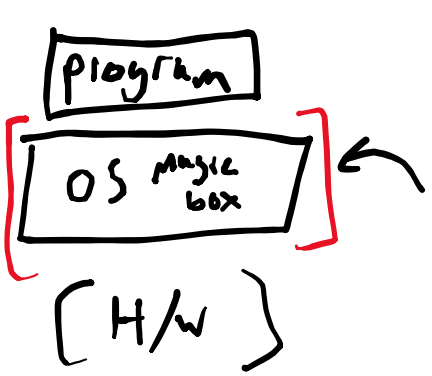
FSCK lab due tonight!

Final exam 2hrs 5/15



CPU reg.

1 CPU look like many private address spce

1. virtualization
2. Concurrency → multiple threads
3. persistence → Fs I/O/disk

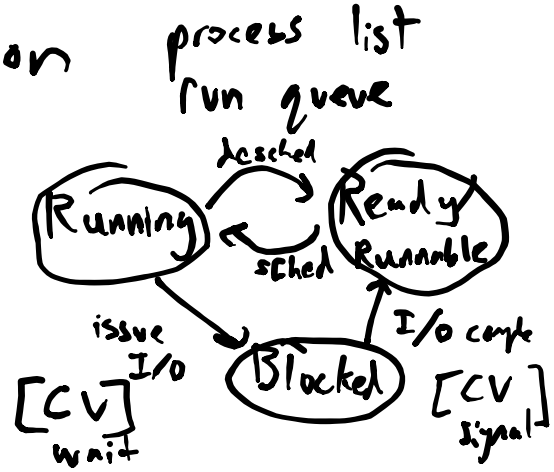
OS makes life easy for programmer

Roles
1. virtual machine (1)
2. standard library (2,3)

[Safety/control + efficiency]

I. Virtualization

process abstraction
fork()
exec()



Limited Direct Execution

privilege levels (user / kernel mode)

trap / return from trap, interrupt

OS @ boot time

initializing trap table

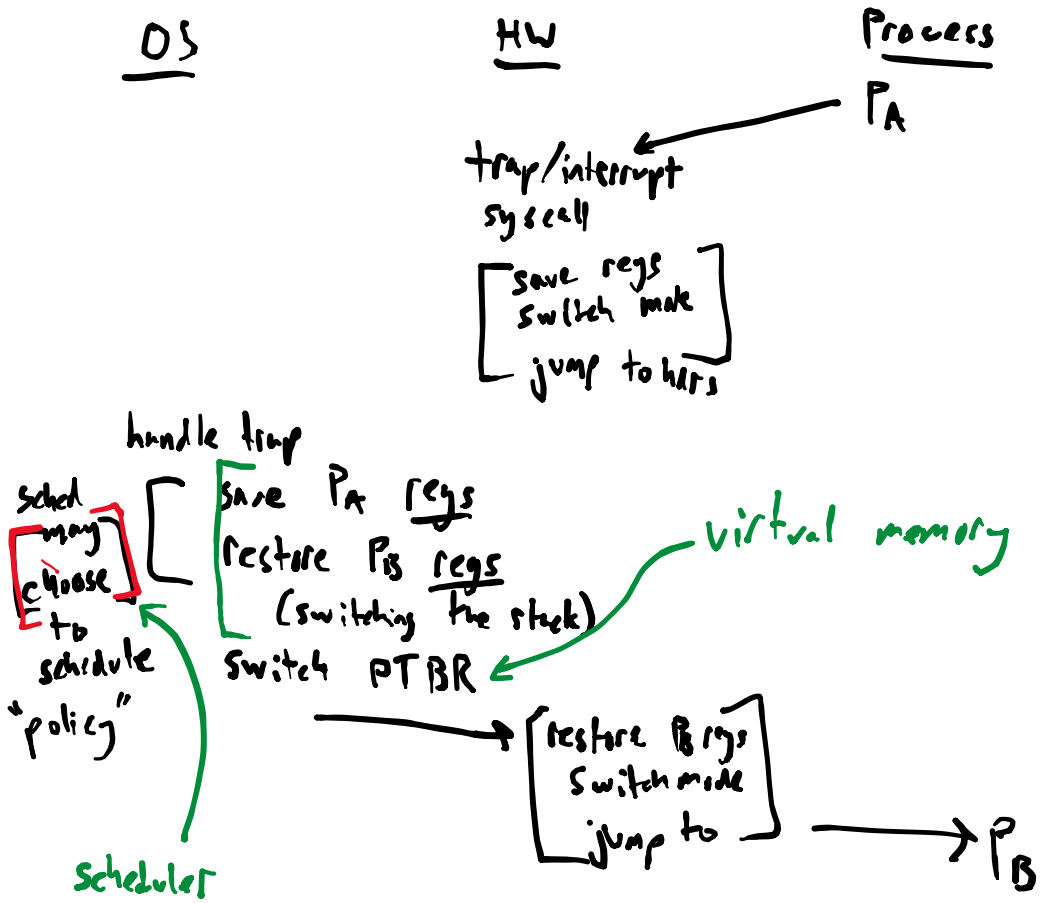
remember addr of
- syscall hdlr
- timer hdlr
- page fault hdlr
- ⋮

start interrupt timer

HW

interrupt X ms

Run time



FIFO, SJF, STCF, RR, MLFQ ← Solaris, BSD Windows

Turnaround time — [batch]

Response time — [interactive]

H : ⋮ L :

Proportional share: Lottery, stride, Linux CFS

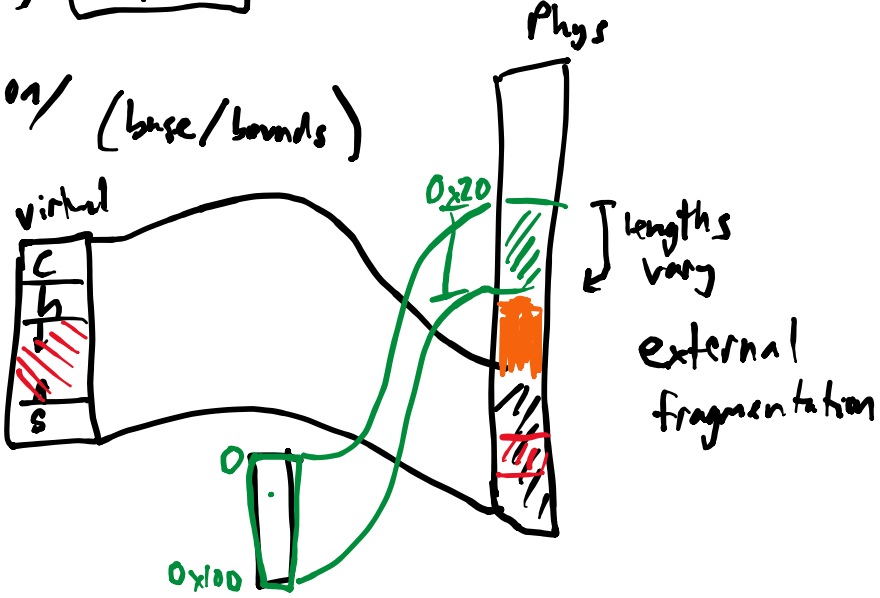
% CPU

Memory management

PA ← VA


```
mov  0x1000, %rl   <- VA
add  1, %rl        <- VA
mov  %rl, 0x1000   <- VA
```

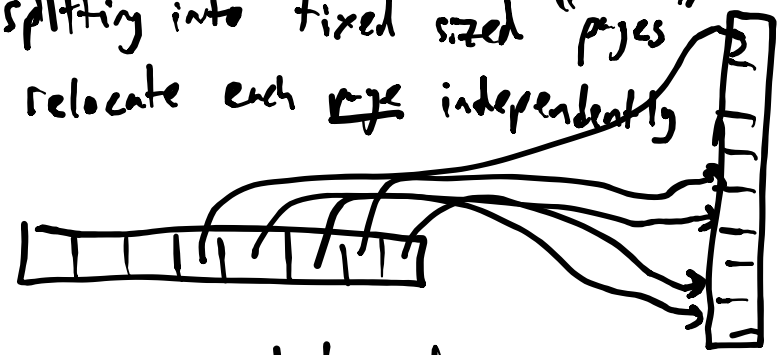
Dynamic Relocation / Segmentation (base/bounds)



+ simple
+ fast
- flexible
- wasted space / fragmentation

Paging: splitting into fixed sized "pages"

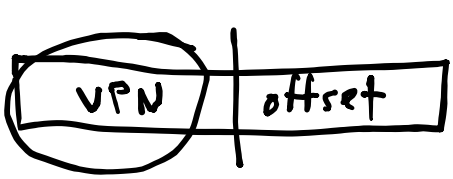
relocate each page independently



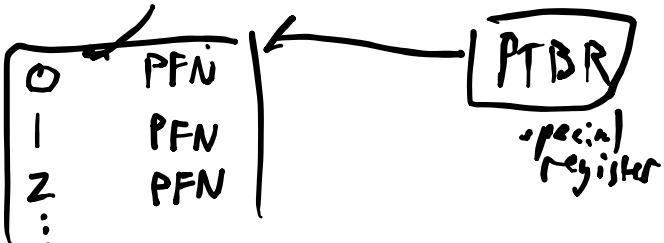
there are lots of mappings!

page table

VPN → PFN



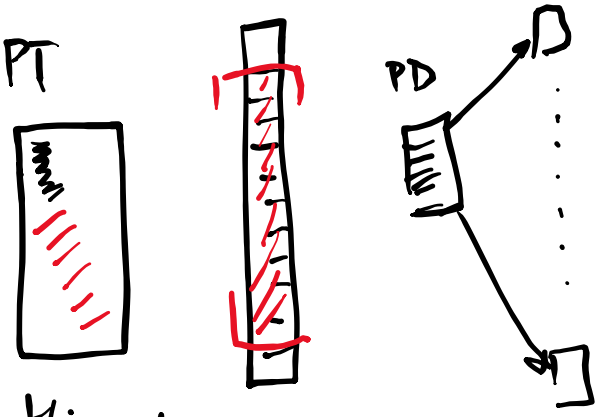
PT (in mem) →



Slow → TLB
address translation cache
H/w or S/w

4-level tree

Size : multi-level PT
large pages



Swapping : if everything doesn't fit
page replacement policies
FIFO
→ OPT
LRU ← least recently used
approximate w/ clock

II Concurrency

multiple threads
↳ share address space
code
heap
globals

1. Need for mutual exclusion
counter increment

atomic

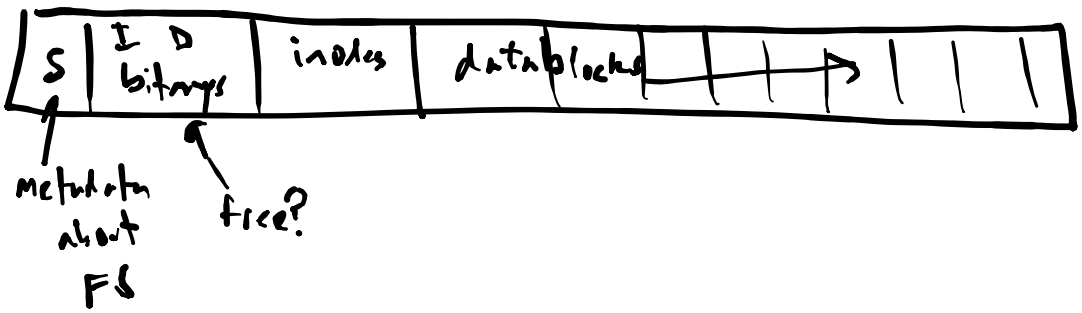
```
mov addr, %r1
inc %r1
mov %r1, addr
```

EVIL scheduler

how to build them from atomic H/w primitives

locks, CVs, semaphores, deadlock

III persistence


S
I D bitmaps
inodes
data blocks
metadata about FS
free?