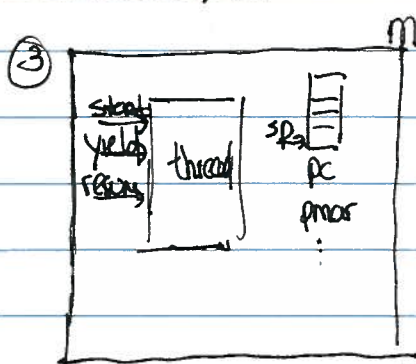


L5 (2/17/09)

① 386 - 86 machine
(review vm slide for x86)
- two ways: segmentation (obsolete)
paging
- 64 bit? x86.

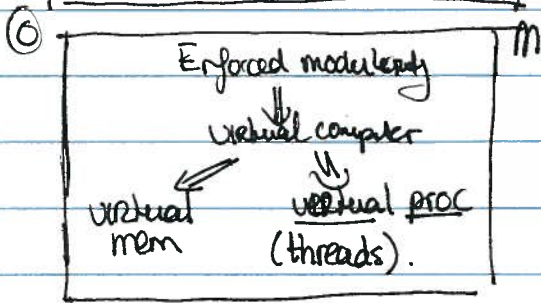


⑤ w. threads

```

1 while (input <= processed) {
2   yield()
3 }

```



④

```

read(fd, buf, 1)
while (input <= processed) {
  processed++
}
sequence coordination

```

polling loop.

⑥

```

yield() {
  save
  schedule
  dispatch
}
int table[7]
int next;

```

① Emacs

init() → wait for input edit() → thread = module in exec

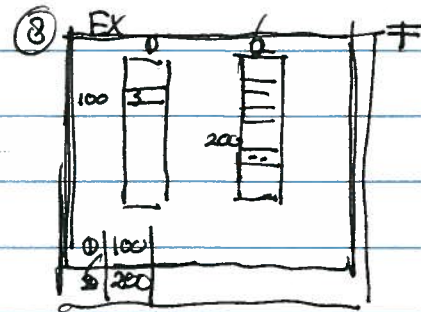
③

```

yield() {
  table[next] ← sp
  n ← (next+1) % 7
  sp ← table[next]
}

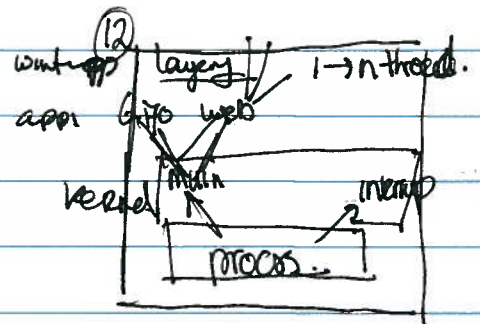
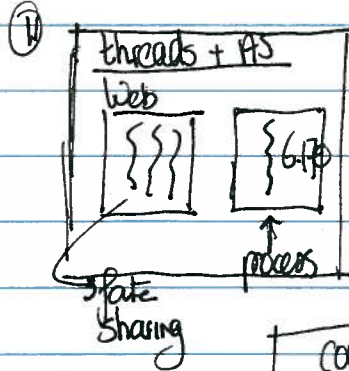
```

(slide)



⑩ thread

- 1) create + destroy
- 2) wait + notify → modify 5 to show
- 3) preemptive → clock interrupt calls yield on behalf of th. (be careful coordinate (Morden))



⑬ which yield?

- asynchronous call + polling / upcall
- lower-level scheduler knows about high
- process.

could talk about switching AS + threads base, kernel the kernel, 3-step atomic process in details perhaps using x86.