Lecture 3:

Scheduling

Last time: CPU virtualization [Fall '11]

Mechanisms: Limited Direct Execution

protocol:
- @ boot: setup trap handlers
- then run jobs:
  - ⇒ OS involved on sys calls (notion of privilege)
  - ⇒ OS involved on timer interrupts (ctxt switch)
- result: efficient, but w/ control [OS as manager]

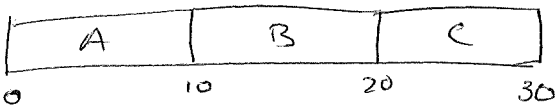
Today: Policies (easier to understand)

based on notion of time sharing (job ~ running program)

Assume: simplistic
- ⇒ set of jobs, all arrive @ once
- ⇒ each just uses CPU (no I/O)
- ⇒ each runs for same length of time
- ⇒ time is known a priori
- ⇒ one metric: turnaround time
  - def: $T_{complete} - T_{arrive}$

Algorithm #1: FIFO / FCFS (queue)

A, B, C get queued up, each run 10 time units

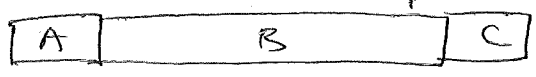


$T_A = 10$ $T_B = 20$ $T_C = 30$ $T_{AVG} = 20$

goal: minimize T_{AVG}

But, Relax #3: jobs of different lengths

Q) how does FIFO do? convoy: C stuck behind B (real life)



$T_{AVG} = 80$

Algorithm #2: SJF ⇒ $T_{AVG} = 50$

optimal (given our assumptions)

But, Relax #1: not all at same time ⇒ problem?

(A: 10, B: 100 arrive @ T=0; C: 10 arrives @ T=20)

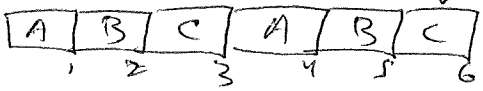
Algorithm #3: STCF (allows preemption)

But, Relax #5: other metrics: Response Time = $T_{first\ run} - T_{arrive}$

Algorithm #4: RR

concept (time slice)

timer interrupt every X ms; & time slice is n·X (n ≥ 1)



R_{AVG} = ?

T_{AVG} = ? (just intuition)

But: what about I/O? (interactive jobs use short bursts of CPU)

More Real Workload:

mix:
- interactive: short CPU bursts, then long pauses (blocked)
- batch: long running CPU workloads - intence

More Real Scheduler:

can't assume knowledge of jobs

⇒ what to do? HISTORY

MLFQ: basic idea:

assume new job is interactive, learn if this is true/not over time!

start w/ RR, but many queues:

- Q_2 → A, B (high pri)
- Q_1 → C
- Q_0 → D, E (low pri)

each job has a priority @ a given time (may change)

Base Rules:

1) if $P(A) > P(B)$ ⇒ A runs
2) if $P(A) == P(B)$ ⇒ RR (A, B)

key: pri changes over time: how?

3) start jobs @ highest priority

Attempt #1:

4) if I/O issued before time slice is up ⇒ (e.g. I/O)
   a) stay @ same pri
   b) if not, move down one level

(EXAMPLES)

long running batch, late arriving interactive

Problems? 1) starvation, job character. 2) Game the scheduler

Attempt #2: solve starvation

Attempt #3: solve gaming

(Also: time slice length)