Fair Share Scheduling

→ From user POV; the concern is about how the application performs

→ Scheduling decisions is based on process sets.

→ Can be extended to a group of users

→ More to those who have less than their fair share (& viceversa)

↳ Not always equal; also weight dependent

Formula

$CPU_j(i)$ = Proc utilization by j through i

$$CPU_j(i) = \frac{CPU_j(i-1)}{2}$$

$GCPU_j(i)$ = Similar, but

for Group

$$GCPU_j(i) = \frac{GCPU_j(i-1)}{2}$$

$$P_j(i) = Base_j + \frac{CPU_j(i)}{2} + \frac{GCPU_k(i)}{4 w_k}$$

w_k = weight of group k ; $\sum_k w_k = 1$

→ If processor is used, priority drops as P_j is increasing

→ If weightage is high; effect on P_j would be low; but as P_j value is also low; the priority is high

Example

→ Count of Proc c is incremented for the

Group column; as the Group is used. Proc count is constant

Traditional Unix Scheduling

→ Integer divison (floor division)

→ Multilevel feedback using Round Robin within each

→ Priority based on the process type + execution history.

Formula

→ $P_j(i) = Base + \frac{CPU_j(i)}{2} + nice$

Summary: Fair Share Scheduling and Priority calculation formula