Project 5 - Fair-Share Scheduler

Purpose

Scheduling is the method by which threads or processes are given access to system resources (e.g. processor time). Scheduling is usually concerned with load balancing a system effectively to achieve a target quality of service. Fair-share scheduling (FSS) is a scheduling strategy for computer operating systems in which the CPU usage is equally distributed among system users or groups, as opposed to equal distribution among processes.

Project Description

For Project 5, you are to add a fair-share scheduler to your OS.

The following are important guidelines for the FSS exercise:

1. Make the following changes:
    - Define (if not already there) a global scheduler mode variable (int scheduler_mode;) to your os345.c such:
        - zero => prioritized round-robin scheduler. (project 2 mode)
        - non-zero => fair scheduler.
    - Add a task time variable to each TCB entry.
2. Fair-share scheduling functionality is defined as follows:
    - Schedule only tasks from the Ready Queue with a non-zero time.
    - Decrement task time by 1 each time the task is scheduled.
    - When all tasks in the Ready Queue have a zero time, then re-compute new "fair" task times.

- Tasks are grouped according to parent task. For example:
    - Level 1:
        - Parent 1 creates 8 child tasks (1/2 time)
        - Parent 2 creates 2 child tasks (1/2 time)
    - Level 2:
        - Parent 1 and each child of Parent 1 would receive 1 clock (for a total of 9 clocks)
        - Parent 2 and each child of Parent 2 would recieve 9/3 = 3 clocks (also for a total of 9 clocks).
- If a parent task remains alive, it should share clocks equally with its children. (Give the parent any fractional clocks. ie. 11/3 = the 2 children get 3 clocks each and the parent receives 5 clocks.)
- Blocked tasks keep and use remaining task time when inserted back into the ready queue.

3. P5 executes the following:
    - ">>P5 0" selects a prioritized round-robin scheduler.
    - ">>P5 1" selects a fair scheduler.
    - ">>P5" schedules 5 new parent tasks and and a printout task that outputs the parent counts every 10 seconds.
    - Each parent task creates a random number of child tasks.
    - Each child task increments a parent counter and swaps.
4. Use ">>P5 0" and ">>P5 1" to dynamically switch between schedulers.

Fair scheduling

Fair-share scheduling is a scheduling strategy for computer operating systems in which the CPU usage is equally distributed among system users or groups, as opposed to equal distribution among processes.

For example, if four users (A,B,C,D) are concurrently executing one process each, the scheduler will logically divide the available CPU cycles such that each user gets 25% of the whole (100% / 4 = 25%). If user B starts a second process, each user will still receive 25% of

the total cycles, but each of user B's processes will now use 12.5%. On the other hand, if a new user starts a process on the system, the scheduler will reapportion the available CPU cycles such that each user gets 20% of the whole (100% / 5 = 20%).

Another layer of abstraction allows us to partition users into groups, and apply the fair share algorithm to the groups as well. In this case, the available CPU cycles are divided first among the groups, then among the users within the groups, and then among the processes for that user. For example, if there are three groups (1,2,3) containing three, two, and four users respectively, the available CPU cycles will be distributed as follows:

100% / 3 groups = 33.3% per group
Group 1: (33.3% / 3 users) = 11.1% per user
Group 2: (33.3% / 2 users) = 16.7% per user
Group 3: (33.3% / 4 users) = 8.3% per user

Grading and Pass-off

Your scheduler is to be demonstrated in person the professor. The assignment is worth 10 points, which will be awarded as follows:

Points	*Requirement*
2	The P5 command dynamically switches between schedulers.
4	Only tasks in the Ready Queue with a non-zero time are scheduled. When all tasks in the Ready Queue have a zero time, then new task times are calculated in a "fair" way.
2	Tasks are grouped according to their parent task. Group counts are approximately the same.
2	Parents share in clock time with their children. Fractional clocks are given to the parent.
-1/2	For each school day late.

In addition, **after completing the above requirements**, the following bonus points may be awarded:

Points	*Requirement*
1	Pass-off lab at least one day before due date.
2	Your fair scheduler works with dead ancestors.