

On Red Black Trees

You need to be alert to (usually minor) changes that may be made to the assignment statement or to the guidelines after the assignment is first put up. Refresh this frame and re-read the assignment carefully before you make your final submission.

In a multi-tasking operating system, a scheduler is a piece of code that decides which process/task shall be given access to the CPU and for how much time. The scheduler maintains a set of pending processes stored in some datastructure S . Everytime the scheduler wakes up, it finds the process with least pending execution and gives it the CPU (this is just one example of a simple scheduling policy). Red Black tree is the structure of choice in implementing scheduler for the linux kernel (2.6.23).

Reasons : It is better to use a balanced search tree for such applications since one would like to minimize the time required for basic operations in the worst case. Like AVL trees, Red Black trees offer $O(\log n)$ insertion, deletion, and search. However, AVL trees are more rigidly balanced than Red Black trees, leading to slower insertion and removal.

Assignment statements

Implement suitable routines for Insertion, Search and Deletion in RB trees. Write a scheduler and process creator which works as follows.

1. In every iteration of an outer loop L , if the total number of live processes is less than some N , the process creator creates a process with execution time between $[1,1000]$ and priority between $[1,4]$.
2. Live processes are stored in a Red Black tree keyed by their pending execution time.
3. In every iteration of an outer loop L , the scheduler
 - i. checks if there is any newly created process and inserts it to the tree,
 - ii. searches the process with least pending execution time and gives it to the CPU, a process with priority i executes for $i \times 50$ seconds once scheduled.
 - iii. Once the process finishes its quota of execution, the scheduler inserts it back to the tree if the process has not completed its entire execution.
4. For insertion of elements in the RB tree whose key value is same as the key of some existing element, put both elements at the same node in a list structure.

Main function :

Write a main function from which the value of N can be set and the system can be simulated for execution and completion of M number of processes. The value of M is also set in `main()`. The output is expected to be a text file containing a table with column headings:

- i. process number,
- ii. creation time,
- iii. priority,
- iv. time stamps when a process got scheduled,
- v. time stamps when a process got preempted from CPU,
- vi. time stamp when a process completed execution.

Submissibles

1. Efficient C program containing the `main()` along with routines for insertion, search, deletion, process scheduler and process creator.
2. A report outlining complexity analysis of red black tree operations.

Marking Guidelines

Assignment marking is to be done only **after** the deadline expires, as submissions gets blocked after the assignment is marked.

Red Black Tree Routines	20
Process Management	20
Report as per assignment statement	10
<i>Total Marks</i>	50

Assignment submission

You need to submit a program file for the assignment and the report and also enhanced/optimised versions of routines (individual submission).

Use electronic submission via the [WBCM link](#)

You should keep submitting your incomplete assignment from time to time after making some progress, as you can submit any number of times before the deadline expires.

Warning

Cases of copying will be dealt with seriously and severely, with recommendation to the Dean to de-register the student from the course.