

1. Accessing a Kernel RB tree via Device File Interface

key words: Linux kernel RB tree, Linux module and device driver, Kprobe, x86's TSC (Time stamp counter) to measure elapse time, Multi-threaded programs.

- Develop a Linux kernel module which initiates an empty RB tree in Linux kernel and allows the tree being accessed as a device file.
- The RB tree is implemented in kernel space as a device "rbt_dev" and managed by a device driver "rbt_drv".
- When the device driver is installed, the tree "rbt" is created and a device "rbt_dev" is added to Linux device file systems.
- The device driver is implemented as a Linux kernel module and enable the file operations: open, write, read, ioctl, and release.

write: if the input object of `rb_object_t` has a non-zero data field, a node is created and added to the `rbt`. If an object with the same key already exists in the tree, it should be replaced with the new one. If the data field is 0, any existing object with the input `key` is deleted from the table.

read: to retrieve the first object (the object with the minimal "key") or last object (the object with the maximal "key") from the RB tree. If the RB tree is empty, -1 is returned and `errno` is set to `EINVAL`. After reading, the object is removed from the tree.

ioctl: The command "set_end" to set which object is to be read. If the argument is 0, read calls retrieve the first object in the tree. If it is 1, read calls get the last object. Otherwise, -1 is returned and `errno` is set to `EINVAL`.

- To test the driver, a user program is developed in which the main program creates 4 threads to populate (by calling write operation) the RB tree with a total of 40 objects and then invoke read and write randomly.
- The threads are set with different real-time priorities and consecutive file operations are invoked after a random delay. After a total of 100 read and write operations are done, the threads should terminate and the main program dumps out all objects in the table.

