

Blatt 4

Rot-Schwarz-Bäume im Linux-Kernel

In der Vorlesung wurden Euch Rot-Schwarz-Bäume konzeptionell und am Pseudo-Code vorgestellt. Des Weiteren wurde deren Nutzung zur Realisierung des Completely Fair Scheduler behandelt.

In diesem Übungszettel soll untersucht werden, wie Rot-Schwarz-Bäume im Linux-Kernel (`include/linux/rbtree.h`) konkret umgesetzt wurden. Beschreibt hierzu die Aufgaben und Felder der Datenstrukturen `struct rb_root` und `struct rb_node`.

Erläutert was für eine Datenstruktur zu tun ist, um sie in einen Rot-Schwarz-Baum einzufügen. Schaut Euch hierzu die Nutzung des Makros `rb_entry` an.

Anschließend sollt Ihr anhand der Methode `__enqueue_entity` des Completely-Fair-Schedulers beschreiben, wie das Einfügen in einen Baum realisiert wurde. Geht dabei auch auf die Rolle der Variablen `link` und `parent` ein. Inwiefern lässt sich der Pseudo-Code aus der Vorlesung in der Realisierung des Kernels wiederfinden? Warum ist das Einfügen in einen Baum nicht in einer komplett generischen Funktion unter Mitgabe eines Comparators gelöst worden? Einen Hinweis hierzu findet Ihr als Kommentar am Anfang der Datei `include/linux/rbtree.h`.

Illustriert Eure Beschreibung mit Code-Schnipseln aus dem Kernel. Der Umfang der Aufgabenlösung sollte sich im Rahmen von 2–3 Seiten bewegen.

Abgabe: Bis Dienstag, 19.06.2012, in der Vorlesung.