

```
// 1
```

```
// Ο "rmp" δείχνει στη διεργασία που μόλις ολοκλήρωσε το κβάντο της
```

```
rmp->proc_usage= rmp->proc_usage+200;
```

```
rmp->grp_usage= rmp->grp_usage+200;
```

```
// 2
```

```
for (proc_nr=0, rmt=schedproc; proc_nr < NR_PROCS; proc_nr++, rmt++) {
```

```
    if (rmt->flags & IN_USE) {
```

```
        if (rmt->procgrp == rmp->procgrp) {
```

```
            rmt->grp_usage=rmp->grp_usage;
```

```
        }
```

```
    }
```

```
}
```

```
// 3
```

```
// Αρχικοποιεί ένα πίνακα B με μηδενικά
```

```
for (proc_nr=0; proc_nr < NR_PROCS; proc_nr++) {
```

```
    B[proc_nr] = 0;
```

```
}
```

```
// Ελέγχει κάθε "procgrp" αν υπάρχει στον πίνακα B. Αν ΔΕΝ υπάρχει, τότε το βάζει
```

```
//στον B 'και αυξάνει τους "hristes" αλλιώς δεν κάνει κάτι. Με τη μεταβλητή "q"
```

```
//διασχίζει τον πίνακα B όταν ελέγχει αν υπάρχει το "procgrp" μέσα στον B.
```

```
hristes=0;
```

```
for (proc_nr=0, rmt=schedproc; proc_nr < NR_PROCS; proc_nr++, rmt++) {
```

```
    if (rmt->flags & IN_USE) {
```

```
        q = 0;
```

```
        while (B[q] != 0)
```

```
        {            if (rmt->procgrp == B[q])
```

```
                        break;
```

```
                    else
```

```
                        q++;
```

```
        }
```

```
        if (B[q] == 0)
```

```
        {            B[q] = rmt->procgrp;
```

```
                    hristes++;
```

```
        }
```

```
    }
```

```
}
```

```
// 4
for (proc_nr=0, rmt=schedproc; proc_nr < NR_PROCS; proc_nr++, rmt++) {
    if (rmt->flags & IN_USE) {
        rmt->proc_usage= rmp->proc_usage/2;
        rmt->grp_usage= rmp->grp_usage/2;
        rmt->fss_priority= rmt->proc_usage/2 + rmt->grp_usage*hristes/4;
        schedule_process_local(rmt);
    }
}
```