

```
// 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, x=schedproc; proc_nr < NR_PROCS; proc_nr++, x++) {
```

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

```
        // Στο “proc_nr_n” είναι στον πίνακα “schedproc” η θέση της
```

```
        //διεργασίας που μόλις ολοκλήρωσε το κβάντο της
```

```
        if (proc_nr != proc_nr_n && x->procgrp == rmp->procgrp) {
```

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

```
        }
```

```
    }
```

```
}
```

```
// 3
```

```
// Στον πίνακα “users” κρατάμε όλα τα “procgrp” που υπάρχουν στον “schedproc”
```

```
used = 0;
```

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

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

```
        users[used] = x->procgrp;
```

```
        used = used + 1;
```

```
    }
```

```
}
```

```
// Ταξινόμηση του πίνακα “users” με ταξινόμηση φυσαλίδας (bubblesort)
```

```
for (i = 1; i <= used-1; i++) {
```

```
    for (j = used-1; j >= i; j--) {
```

```
        if (users[j-1] > users[j]) {
```

```
            temp = users[j-1];
```

```
            users[j-1] = users[j];
```

```
            users[j] = temp;
```

```
        }
```

```
    }
```

```
}
```

```

// Διάσχιση του ΠΛΕΟΝ ταξινομημένου πίνακα "users" για να βρω το πλήθος των
//διαφορετικών "groups" σε αυτή τη μεταβλητή
groups = 1;

for (i = 1; i <= used-1; i++) {

    if (users[i-1] < users[i]) {

        groups = groups + 1;

    }

}

// 4

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

    if (x->flags & IN_USE) {

        x->proc_usage = x->proc_usage/2;

        x->grp_usage = x->grp_usage/2;

        x->fss_priority = x->proc_usage/2 + x->grp_usage*groups/4 + 0;

        schedule_process_local(x);

    }

}

```