

```
// 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
```

```
d=schedproc;
```

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

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

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

```
            d->grp_usage=rmp->grp_usage;
```

```
        }
```

```
    }
```

```
    d++;
```

```
}
```

```
// 4 (χωρίς τον τύπο για fss_priority γιατί χρειάζεται το “number_of_groups”)
```

```
d=schedproc;
```

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

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

```
        d->proc_usage = d->proc_usage/2;
```

```
        d->grp_usage = d->grp_usage/2;
```

```
    }
```

```
    d++;
```

```
}
```

```

// 3
// Στη μεταβλητή "new_group" βάζει 0. Για κάθε θέση του πίνακα
// "schedproc" (δείκτης "d") ελέγχει από την επόμενη θέση (δείκτης "in_d=d+1;")
// μέχρι το τέλος του πίνακα "schedproc". Αν βρει ΙΔΙΟ "procgrp" μπαίνει στο else
// και βάζει new_group=1; , που σημαίνει το ξαναβρήκα. Μόνο αν το
// "new_group==0" σημαίνει ότι βρήκε ΝΕΟ χρήστη, άρα αυξάνει το "nog++"

nog=0; // αριθμός ομάδων "number_of_groups"

d=schedproc;

for (y=0; y<NR_PROCS; y++) {

    if (d->flags & IN_USE) {

        in_d=d+1;

        new_group=0;

        for (in_y=y+1; in_y<NR_PROCS; in_y++) {

            if (in_d->flags & IN_USE) {

                if (in_d->procgrp!=d->procgrp) {

                    in_d++;

                    continue;

                }

                else {

                    new_group=1;

                    break;

                }

            }

            in_d++;

        }

        if (new_group==0) {

            nog++;

        }

    }

    d++;

}

```

```
// 4 (για fss_priority, τώρα έχω βρει το "nog" - number_of_groups)

base=0;

d=schedproc;

for (y=0; y < NR_PROCS; y++) {

    if (d->flags & IN_USE) {

        d->fss_priority = d->proc_usage/2 + d->grp_usage*nog/4 + base;

        schedule_process_local(d);

    }

    d++;

}
```