

# Descripció Workloads:

## CPU:

Consisteix en un doble bucle for de 32000\*2048 iteracions amb una comparació a dins i una suma.

```
for (j = 1; j < 32000; ++j) {  
    for (k = 1; k < 2048; ++k) {  
        if (j % 2 == 0 && k % 2 == 0) ++useless;  
    }  
}
```

## I/O:

Consisteix en inicialitzar i recórrer un vector de ints de 1024 iteracions i fer una suma. Després es bloqueja per 10 ticks i en acabar de recórrer el vector es bloqueja 500 ticks.

```
useless = 0;  
int vector[1024];  
for (j = 0; j < 1024; ++j) {  
    vector[j] = j;  
    useless += vector[j];  
    read(0,buff,10);  
}  
read(0,buff,500);
```

## Mix:

Es tracta d'executar els dos codis anteriors un seguit del altre.

## Conclusions:

Els resultats estan en el fitxer soa\_schedPerf.xls i les nostres conclusions són les següents:

### CPU:

Quan només executem el workload de CPU en el cas del RoundRobin els tres processos triguen el mateix en executar i estan en ready més o menys el mateix temps perquè tenen el mateix quantum i el planificador els dona la mateixa prioritat i temps de cpu a tots. Quan se li acaba el quantum a un procés executa un altre.

En el cas del FCFS el que passa és que com que en el workload no hi ha cap bloqueig per I/O, s'executa primer tot el procés 1, mentre els processos 2 i 3 esperen en cua de ready. Quan el procés 1 acaba executa el procés 2 i l'últim espera. El procés idle mai executa perquè sempre hi ha algun procés executant.

### I/O:

En el cas del RoundRobin tenim poc a comentar perquè no sabem quant temps estan en blocked els processos 1, 2 i 3, però sí que podem veure que el procés idle executa bastanta estona degut a que els 3 processos estan bloquejats.

En el cas del FCFS veiem com en el cas del I/O la carrega entre els processos és semblant, doncs quan es bloqueja un procés immediatament després executa un altre. Podem veure com el idle executa molts ticks, corresponent al període de temps en que els 3 processos estan bloquejats.

### Mix:

En el cas del MIX en ambdós casos podem veure que el resultat és pràcticament el mateix que sumant els workloads de CPU i I/O.

Així doncs, veiem com en el cas del RoundRobin els processos tenen una carrega semblant en tot menys en System, i que en el FCFS desapareix el problema del workload 1 dels processos en ready perquè com també fem I/O els processos es van executant de manera no seqüencial.

No podem fer una valoració global per que ens falten els ticks de blocked en RoundRobin.