

SISTEMAS CONCURRENTE Y DISTRIBUIDOS.

Práctica 4. Sistemas de tiempo real.

Pablo Manresa Nebot. 2ºA2

Índice

Ejecutivo1-compr	1
Ejecutivo2	2

Ejecutivo1-compr

Cada vez que acaba un ciclo secundario, se informa del retraso del instante final actual respecto al instante final esperado. Si se comprueba que dicho retraso es superior a 20 milisegundos, el programa aborta con un mensaje de error.

Para ello, se ha realizado lo siguiente:

```
    fin_sec = steady_clock::now();

    milliseconds_f retraso = fin_sec - ini_sec;

    cout << "RETRASO : " << retraso.count() << endl;

    if(retraso.count() > milliseconds(20).count() ){
        cout << "\n Abortar programa. Retraso > 20 milisegundos " <<
retraso.count() << "\n";
        exit(EXIT_FAILURE);
    }
```

Justo después de:

```
    ini_sec += Ts ;
    sleep_until( ini_sec );
```

Para verificar si el retraso producido excede o no los 20 milisegundos.

Ejecutivo2

Diseño de planificación de tareas y restricciones que cumplan con lo siguiente:

Tarea	T	C
A	500	100
B	500	150
C	1000	200
D	2000	240

La solución ha sido la siguiente:

```
// Datos de las tareas:
// -----
// Ta.  T    C
// -----
// A  500  100
// B  500  150
// C 1000  200
// D 2000  240
// -----
//
// Planificación (con Ts == 500 ms)
// *-----*-----*-----*
// | A B C   | A B D     | A B C   | A B     |
// *-----*-----*-----*-----*
```

→ **¿ cual es el mínimo tiempo de espera que queda al final de las iteraciones del ciclo secundario con tu solución ?**

El mínimo tiempo de espera corresponde al segundo ciclo, que contiene los procesos A B D, que juntos suman 490 ms. , siendo 500 ms. el tiempo máximo. Por lo tanto, corresponde a una espera de 10 ms.

→ **¿ sería planificable si la tarea D tuviese un tiempo cómputo de 250 ms ?**

Sí, porque justo sumaría el ciclo A B D un total de 500 ms. siendo la espera de 0 ms. por lo que, sería planificable.