



Sistemas Operativos

Sincronización de Threads

Diego Madariaga

1.

Impresora Compartida

Impresora compartida

Suponga que se dispone de una impresora compartida por múltiples threads. Dos o más threads pueden usar simultáneamente la impresora, pero en este caso las líneas de la impresora saldrían entremezcladas.

En una aplicación, se desea evitar el entremezclado de líneas haciendo que cada thread solicite el acceso exclusivo a la impresora antes de ocuparla y además notifique cuando termina de utilizarla. Por lo tanto, un thread tendrá la siguiente forma:

```
tarea() {  
    ...  
    obtenerImpresora();  
    ... /* utilizar impresora */  
    devolverImpresora();  
    ...  
}
```

Impresora compartida

Además, por razones de ahorro de electricidad, se necesita que la impresora se coloque en modo bajo consumo cada vez que transcurran 5 minutos sin ser utilizada por ninguna tarea. Para colocar la impresora en modo bajo consumo invoque el procedimiento `modoBajoConsumo()`. Para volver a usar la impresora cuando está en modo bajo consumo, invoque `modoUsoNormal()`.

Implemente `obtenerImpresora()`, `devolverImpresora()` e `inicializarImpresora()`.

2.

Función *más Parecidas*

Función másParecidas

La función **masParecidas**, encuentra las 2 tareas que más se parecen. Esta función recibe como parámetros un arreglo tareas con las **n** tareas de los alumnos y 2 punteros **pi** y **pj** con la direcciones de las variables en donde se deben almacenar los índices de las 2 tareas más parecidas en el arreglo.

```
void masParecidas(Tarea *tareas, int n, int *pi, int *pj) {
    int min= MAXINT;
    for (int i= 1; i<n; i++)
        for (int j= 0; j<i; j++) {
            int similitud= compTareas(tareas[i], tareas[j]);
            if (similitud<min) {
                min= similitud;
                *pi= i; *pj= j;
            }
        }
    }
}
```

Función más Parecidas

La función `compTareas` es dada y entrega un coeficiente de similitud. El 0 significa que son iguales. Para comparar 2 tareas esta función toma tiempo muy variable: desde unos pocos milisegundos a minutos. Las $O(n^{**2})$ comparaciones toman mucho tiempo de CPU.

Paralelice la función `masParecidas` considerando una máquina octa-core. No se pueden ejecutar más de 8 comparaciones de tareas en paralelo porque se requeriría mucha memoria. Un core no puede permanecer ocioso mientras queden comparaciones por realizar.

Función más Parecidas

- ▷ Patrón de paralelización de productor/consumidor
- ▷ El productor genera Jobs con órdenes de comparación de un par de tareas
- ▷ Los múltiples consumidores extraen los jobs para realizar una comparación



Sistemas Operativos

Sincronización de Threads

Diego Madariaga