

**CC41B Sistemas Operativos**  
**Control 1 – Semestre Primavera 2006**  
**Prof.: Luis Mateu**

**Pregunta 1**

(a) Se propone la siguiente implementación para un semáforo:

<pre>int sc= 0, wc= 0;  void signal() {     sc++; }</pre>	<pre>void wait() {     while (wc&lt;=sc)         ;     wc++; }</pre>
---	--

Para los siguientes casos, diga cuando esta implementación es correcta o muestre mediante un diagrama de threads que es incorrecta:

- Un solo thread invoca `signal` y otro thread invoca `wait`.
- Un proceso pesado invoca `signal` y otro proceso pesado invoca `wait`.
- Varios threads pueden invocar tanto `signal` como `wait`.

(Estos puntos tienen distinta ponderación.)

(b) Implemente el semáforo de la parte (a) usando los mensajes de `nSystem`.

**Pregunta 2**

La ATP Tour ha organizado un tipo de torneo de tenis que se caracteriza por durar un solo día. Para ello los partidos son mucho más cortos, con el objeto de jugar un cuadro clásico de tenis en una sola jornada. Actualmente existe una solución secuencial escrita en `nSystem`:

```
char *torneo(char **players, int n) { // n: potencia de 2
    char **draw= (char**)nMalloc(n*2*sizeof(char*));
    int k, r;
    for (k= 0; k<n; k++) draw[n+k]= players[k];
    for (r= n; r>1; r= r/2) {
        /* ronda de r jugadores */
        for (k= r/2; k<r; k++)
            draw[k]= play(draw[2*k], draw[2*k+1]); /* muy lento */
    }
    return draw[1]; /* entrega el ganador del torneo */
}
```

En donde `players` es un arreglo con los nombres de los `n` jugadores que participan en el torneo. El procedimiento `play` es dado y toma mucho tiempo en enfrentar 2 jugadores, cuyos nombres se reciben como parámetros. Este procedimiento retorna el nombre del ganador. El problema es que como se juega un solo partido a la vez, el torneo termina usualmente en la madrugada.

- i. Suponiendo que hay suficientes canchas disponibles, se le pide a Ud. que reescriba el procedimiento `torneo` de tal forma que se jueguen los partidos simultáneamente. En caso de requerir herramientas de sincronización use los monitores de `nSystem`. Ud. debe ser eficiente: si el ganador de Moya y Massu se debe enfrentar con el ganador de Gonzalez y Federer, el partido debe comenzar de inmediato una vez que terminen ambos partidos, ¡y no antes! (es decir que el partido entre `draw[2*k]` y `draw[2*k+1]` debe comenzar en cuanto se conozcan los nombres que van en esas dos posiciones del `draw`).
- ii. Suponga ahora que hay solo 4 canchas disponibles, numeradas de 1 a 4. Modifique `torneo` de modo que ahora se utilice `char *play(char *jug1, char *jug2, int cancha)`, que indica en qué cancha se jugará el partido. Desde luego, Ud. no puede usar una misma cancha para dos partidos simultáneamente. Para la sincronización Ud. debe usar los monitores de `nSystem`.