

**CC41B Sistemas Operativos**  
Tarea 1 – Semestre Primavera 2007  
Prof.: Luis Mateu

La federación de tenis de mesa de Chile organiza un torneo en donde los jugadores se enfrentan todos con todos. En el torneo participan 8 jugadores, numerados de 0 a 7, que se enfrentan en 7 rondas de 4 partidos. Para disminuir la duración del torneo, se requiere que los partidos se jueguen en paralelo, sujeto a que se dispone de N mesas. Para hacer enfrentarse los jugadores x e y en la mesa m se le da a Ud. el procedimiento `enfrentar(x, y, m)`. Este procedimiento toma mucho tiempo y solo retorna cuando el partido finalizó. Algunos partidos pueden ser muy cortos (3 sets) mientras otros pueden ser extensos (5 sets muy disputados).

La siguiente es una solución incorrecta para realizar el torneo pedido:

```
struct { int x, y; } partidos[4*7]= /* 7 fechas, 4 partidos por ronda */
{ {0,4}, {1,5}, {2,6}, {3,7}, /* 0 enfrenta a 4, 1 a 5, etc. por la 1era. ronda, */
  {0,1}, {2,4}, {3,5}, {7,6}, /* 0 enfrenta a 1, 2 a 4, etc. por la 2da. ronda, */
  {0,2}, {3,1}, {7,4}, {6,5}, /* 3era. ronda ... y así hasta la 7ma. */
  {0,3}, {7,2}, {6,1}, {5,4},
  {0,7}, {6,3}, {5,2}, {4,1},
  {0,6}, {5,7}, {4,3}, {1,2},
  {0,5}, {4,6}, {1,7}, {2,3} };
int prox; /* índice del próximo partido que se debe jugar */

void torneo(int N) { /* N<=4 */
  nTask tasks[4];
  int m;
  prox= 0;
  for (m=0; m<N; m++)
    tasks[m]= nEmitTask(mesa, m);
  for (m=0; m<N; m++)
    nWaitTask(tasks[m]);
}

int mesa(int m) {
  while (prox<4*7) {
    int x= partidos[prox].x;
    int y= partidos[prox].y;
    prox++;
    enfrentar(x, y, m);
  }
  return 0;
}
```

Escriba una solución correcta y eficiente de este problema. La solución debe funcionar bajo nSystem y debe utilizar monitores para la sincronización entre las tareas. Por solución correcta y eficiente se entiende que:

- En una mesa no se pueden jugar dos o más partidos simultáneamente.
- Un jugador no puede estar jugando dos o más partidos simultáneamente.
- El procedimiento torneo debe retornar en algún momento y sólo lo hace una vez que todos los partidos finalizaron.
- Ud. debe invocar nWaitTask para cada una de las tareas adicionales que cree.
- Para obtener un avance ordenado del torneo ningún partido de la ronda i+1 podrá comenzar mientras quede algún partido de la ronda i sin comenzar.
- Los partidos de la primera ronda deben comenzar en el mismo orden en que aparecen en el arreglo partidos.
- Los partidos que no violen ninguna de las restricciones anteriores *se deben* ejecutar en paralelo.

### Recursos

En <http://www.dcc.uchile.cl/~lmateu/CC41B> se publicará próximamente el programa de prueba de esta tarea. Este programa le dirá si su tarea funciona y cuan eficiente es.

### Plazo de entrega

La tarea se entrega *funcionando* en U-cursos. Para ello entregue un archivo tar comprimido con gzip que incluya todos los archivos que Ud. creó o modificó. *No incluya archivos binarios*. El plazo de entrega vence el Lunes 11 de Septiembre a la medianoche. Se descontará medio punto por día hábil de atraso. Se recomienda vivamente resolver esta tarea antes del control 1.