

Pregunta 1

Parte a.- Los 5 filósofos decidieron juntarse a comer un asado (una sola vez), pero nadie puede comenzar a comer mientras no hayan llegado todos al asado. El código de cada filósofo es:

```
int filosofo(int i) {
    pensar();
    encuentro(i);
    comerAsado();
    return 0;
}
```

Se proponen estas 2 soluciones para la función *encuentro*:

Solución 1	Solución 2
<pre>int cnt= 0; nSem sem; // = nMakeSem(0); void encuentro(int i) { cnt++; if (cnt==5) { for (int k=0; k<4; k++) nSignalSem(sem); } else nWaitSem(sem); }</pre>	<pre>int flag[5]= { 0, 0, 0, 0, 0}; void encuentro(int i) { flag[i]= 1; for (int k= 0; k<5; k++) { while (!flag[i]) ; } }</pre>

Discuta sobre la correctitud de ambas soluciones (no su eficiencia). Si en una solución es posible hacer un pequeño cambio (en no más de una línea) para que sea correcta, indíquelo. De otra forma haga un diagrama de threads que muestre una situación en donde el resultado es incorrecto.

Parte b.- Programe una solución de *encuentro* que sea correcta y eficiente usando 2 semáforos de nSystem.

Pregunta 2

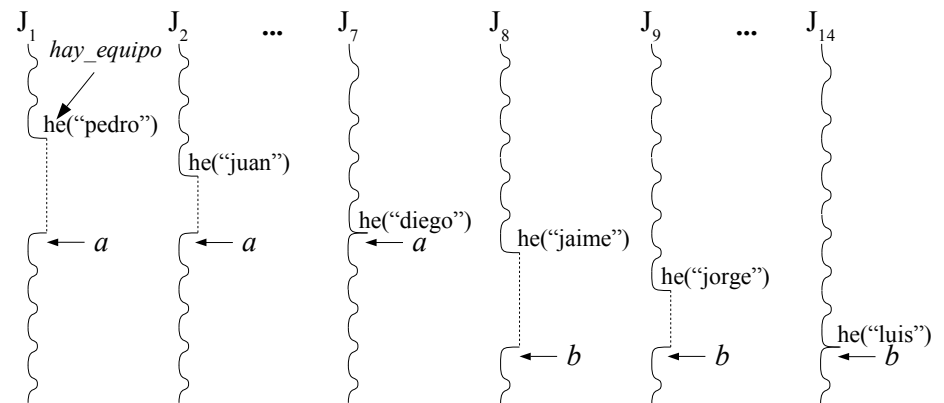
Se necesita formar equipos de 7 jugadores de futbolito. Los jugadores son representados por threads que invocan la función *hay_equipo* indicando como argumento su nombre. Esta función espera hasta que 7 jugadores hayan invocado *hay_equipo* retornando un arreglo de 7 strings con los nombres del equipo completo. Este es un ejemplo del código de un jugador:

```
int jugador(char *nombre) {
    for (;;) {
        dormir();
        char **equipo= hay_equipo(nombre);
        jugar_futbolito(equipo);
        beber_cerveza();
    }
}
```

Programa la función *hay_equipo*. El encabezado es el siguiente:

```
char **hay_equipo(char *nombre);
```

Las funciones *dormir*, *jugar_futbolito* y *beber_cerveza* son dadas. El siguiente diagrama muestra un ejemplo de ejecución:



Observe que la llamada a *hay_equipo* espera hasta que se haya formado un equipo con 7 jugadores. Los primeros 7 jugadores (J₁ a J₇) forman el equipo *a* y por lo tanto sus llamadas a *hay_equipo* retornan el mismo arreglo *a* con los 7 nombres del equipo: “pedro”, “juan”, ..., “diego”. Los siguientes 7 jugadores (J₈ a J₁₄) forman el equipo *b* y sus llamadas a *hay_equipo* retornan el arreglo *b*, distinto de *a*, con los nombres “jaime”, “jorge”, ..., “luis”. Resuelva el problema usando los monitores de nSystem. *Tenga especial cuidado en evitar dataraces asociadas al equipo retornado.*