

## CC3301 Programación de Software de Sistemas

### Tarea 3 – Semestre Primavera 2016 – Prof.: Luis Mateu

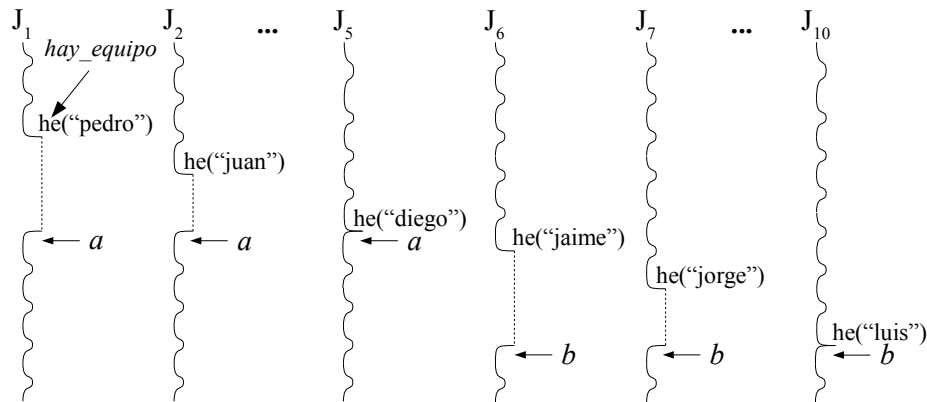
Se necesita formar equipos de 5 jugadores de baby-fútbol. Los jugadores son representados por threads que invocan la función *hay\_equipo* indicando como argumento su nombre. Esta función espera hasta que 5 jugadores hayan invocado la misma función retornando un arreglo de 5 strings con los nombres del equipo completo. Este es un ejemplo del código de un jugador:

```
void *jugador(void *ptr) {
    char *nombre= ptr;
    for (;;) {
        dormir();
        char **equipo= hay_equipo(nombre);
        jugar_baby(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 5 jugadores. Los primeros 5 jugadores (J<sub>1</sub> a J<sub>5</sub>) forman el equipo *a* y por lo tanto sus llamadas a *hay\_equipo* retornan el mismo arreglo *a* con los 5 nombres del equipo: “pedro”, “juan”, ..., “diego”. Los siguientes 5 jugadores (J<sub>6</sub> a J<sub>10</sub>) 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 variables globales como un mutex, una condición, etc.

### Recursos

Baje *t3.zip* de material docente en U-cursos y descomprímalo. El directorio *T3* contiene los archivos *test-equipo.c* que prueba si su tarea funciona, *Makefile* que le servirá para compilar su tarea y *equipo.h* que contiene el encabezado de la función *hay\_equipo*. Ud. debe programar *hay\_equipo* en el archivo *equipo.c*. El programa de prueba lo felicitará si su tarea aprueba todos los tests o le indicará cuál test falla.

### Entrega

Entregue su tarea solo si compila sin arrojar warnings en la máquina *anakena.dcc.uchile.cl* y pasa exitosamente al menos el primer test en *anakena*. El segundo test podría fallar en *anakena* porque esta máquina podría no permitir crear los 100 threads que este test requiere. Pero en su computador personal debe pasar todos los tests, pues durante la corrección su tarea se probará en una máquina que sí permite crear 100 threads.

Se inspeccionará su tarea para verificar que (i) usa correctamente los mutex y condiciones y (ii) no hace busy-waiting.

Ud. debe entregar el archivo *equipo.c* por medio de U-cursos. El plazo de entrega es el Lunes 5 de diciembre a las 23:59. Se descontará medio punto por día de atraso. No se consideran los días sábado, domingo o festivos.