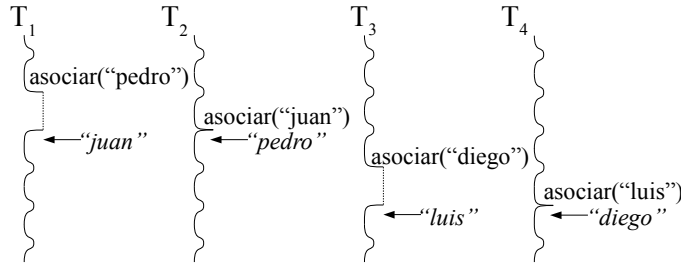


Pregunta 1

La función *asociar* debe obtener el nombre de un socio único. Recibe como argumento el nombre de la persona que busca socio y debe retornar el nombre de su socio. Al llamarla pueden ocurrir solo 2 casos: (i) no hay un socio pendiente, en cuyo caso el llamador queda pendiente hasta que otro thread invoque *asociar*, o (ii) hay un socio pendiente, en cuyo caso el socio del llamador es el socio pendiente y viceversa, y no queda ningún socio pendiente. El siguiente diagrama muestra un ejemplo de ejecución:



La siguiente solución es incorrecta aunque funciona casi siempre:

```

nSem s; // = nMakeSem(0); ¡0 tickets iniciales!
char **psocio= NULL;

char *asociar(char *nom) {
    if (psocio==NULL) {
        psocio= &nom; // no hay un socio pendiente
        nWaitSem(s); // nom queda pendiente a la espera de un socio
    }
    else {
        char *socio; // sí hay un socio pendiente
        socio= *psocio; // el nombre del socio pendiente
        *psocio= nom; // cambia el parámetro nom en el thread pendiente
        nom= socio; // el nombre del socio de este thread
        psocio= NULL; // ahora no hay ningún socio pendiente
        nSignalSem(s); // reanuda al socio
    }
    return nom; // entrega el nombre del socio
}
    
```

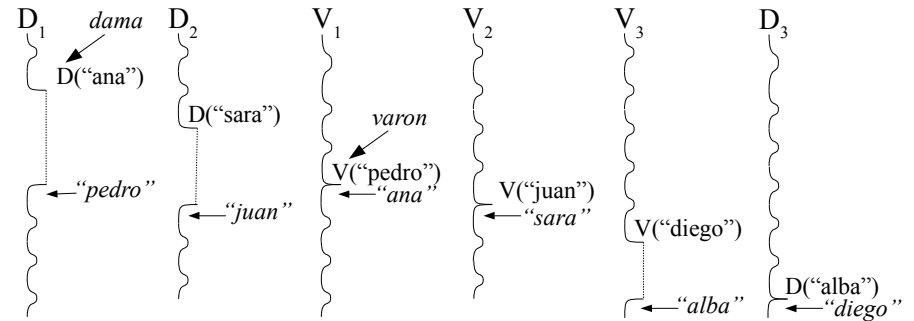
a.- Haga un diagrama de threads que muestre que *asociar*("pedro") podría retornar incorrectamente el mismo "pedro".

b.- Corrija esta solución usando uno o más semáforos adicionales. No olvide indicar los tickets iniciales. No puede usar monitores. Ayuda: lo único incorrecto de la solución de más arriba es el aspecto

sincronización.

Pregunta 2

Las tareas de nSystem representan damas y varones que buscan una pareja para bailar en una discoteca invocando *dama* y *varon*. La función *dama* recibe como parámetro el nombre de la dama. Si hay varones esperando pareja, *dama* entrega de inmediato el nombre del varón que llegó primero. Es decir que la asignación de parejas se hace por orden de llegada. Si no hay varones en espera, *dama* espera por una invocación de la función *varon*. Análogamente *varon* entrega al instante el nombre de la primera dama que todavía busca su pareja, o espera una invocación de *dama*. El siguiente es un diagrama de tareas que muestra un ejemplo de ejecución.



El diagrama muestra que la pareja de ana (D₁) es pedro (V₁) y que por lo tanto *dama* en D₁ retorna "pedro" y *varon* en V₁ retorna "ana".

Implemente las funciones *dama* y *varon* usando los monitores y condiciones de nSystem. Los encabezados de las funciones pedidas son:

```

char *dama(char *nombre);
char *varon(char *nombre);
    
```

Restricciones: Debe evitar cambios de contexto innecesarios. Para la sincronización use un solo monitor de nSystem con múltiples condiciones. Considere que si hay múltiples *nWaitCondition* esperando en la misma condición, un *nSignalCondition* despierta a cualquiera de ellos. Además *nWaitCondition* se podría despertar sin un *nSignalCondition* (como las condiciones de pthreads). Para garantizar la atención por orden de llegada use una *FifoQueue* para las damas en espera y otra *FifoQueue* para los varones en espera.