

CC3301 Programación de Software de Sistemas

Tarea 3 – Semestre Otoño 2017 – Prof.: Luis Mateu

El sabio chino, dueño del restaurant que visitan los 5 filósofos, ha accedido a que un filósofo se siente en cualquiera de las 5 sillas de la mesa. Ya no es necesario que el filósofo i se siente en la i -ésima silla. Con esto la función que representa a cada filósofo queda como sigue:

```
void filosofo(int i) {
    for (;;) {
        int k= buscarSilla(); // k entre 0 y 4
        comer(k, (k+1)%5);
        desocuparSilla(k);
        pensar();
    }
}
```

Programa las funciones *buscarSilla* y *desocuparSilla*. Recuerde que hay solo 5 palitos, que un filósofo que se sienta en la silla k (con k entre 0 y 4) necesita los palitos k y $(k+1)\%5$ para poder comer y que 2 filósofos no pueden comer con el mismo palito simultáneamente. Su solución debe usar eficientemente los palitos, lo que significa que cuando existen sillas con sus 2 palitos desocupados y hay un filósofo con la intención de comer, *buscarSilla* le debe asignar una silla de inmediato. No se requiere evitar hambruna.

Recursos

Baje *t3.zip* de material docente en U-cursos y descomprímalo. El directorio *T3* contiene los archivos *test-silla.c* que prueba si su tarea funciona, *Makefile* que le servirá para compilar su tarea y *silla.h* que contiene los encabezados de las funciones pedidas. Ud. debe programar *buscarSilla* y *desocuparSilla* en el archivo *silla.c*. El programa de prueba lo felicitará si su tarea aprueba todos los tests o le indicará cuál test falla.

Restricciones

Entregue su tarea solo si compila sin arrojar warnings en la máquina *anakena.dcc.uchile.cl* y pasa exitosamente hasta el test de robustez con 5 filósofos en *anakena*. El test de robustez con 50 filósofos podría fallar en *anakena* porque esta máquina no permite crear los 50 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 50 threads.

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

Entrega

Ud. debe entregar el archivo *silla.c* por medio de U-cursos. No modifique *silla.h* o *Makefile* porque esto podría hacer que su tarea no compile durante la corrección. El plazo de entrega es el Lunes 5 de junio a las 23:59. Se descontará medio punto por día de atraso. No se consideran los días sábado, domingo o festivos.