

CC3301 Programación de Software de Sistemas – Tarea 5 – Semestre Primavera 2019 – Prof.: Luis Mateu

En esta tarea Ud. deberá programar un sistema que permita imprimir asincrónicamente documentos en múltiples impresoras. Un thread productor de documentos solicita la impresión de *doc* de la siguiente manera:

```
Doc doc= ...; /* un documento */
Orden *ord= imprimir_async(doc);
... realiza otras tareas ...
confirmar(ord);
...
```

La función *imprimir_async* se usa para solicitar la impresión asíncrona de un documento, es decir sin esperar a que se termine de imprimir. Esta función entrega una orden de trabajo *ord* que es usada más tarde para esperar hasta que el documento sea impreso. Para ello se invoca la función *confirmar* pasando como argumento la orden entregada por *imprimir_async*.

El problema consiste en programar las funciones *imprimir_async* y *confirmar* de tal forma que los documentos se impriman en múltiples impresoras en paralelo. Para realizar efectivamente la impresión de *doc* se dispone de la función *imprimir(doc, impr)*, en donde *impr* es la impresora que se usará. La asignación de los trabajos de impresión a las distintas impresoras se debe hacer de manera que en ningún momento se asignen 2 o más trabajos a la misma impresora simultáneamente. Además una impresora no puede permanecer ociosa habiendo documentos pendientes para imprimir.

Requerimientos

Concretamente se pide programar las siguientes funciones en un archivo con nombre *impr.c* (ver encabezados en archivo *impr.h*):

```
void init_impr(Impr **imprs, int n_impr);
void terminar_impr();
Orden *imprimir_async(Doc *doc);
void confirmar(Orden *ord);
```

Además Ud. debe definir en *impr.c* la estructura con etiqueta *orden* (que corresponde al tipo *Orden*):

```
struct orden { ... };
```

Coloque ahí los campos que Ud. estime necesarios para implementar las funciones pedidas. No redefina el tipo *Orden* con typedef, ya que el typedef para *Orden* está hecho en *impr.h*.

La función *init_impr* es llamada para establecer que *imprs[0]*, *imprs[1]*, ..., etc. son las *n_impr* impresoras disponibles. En esta función cree *n_impr* threads consumidores de documentos, un thread para cada impresora. En cada thread consumidor extraiga una orden de una cola de impresión e invoque la función dada *imprimir* para realizar la impresión efectiva del documento

contenido en la orden. Hágalo en la impresora atribuida a ese thread. Al terminar marque la orden como impresa. La función *imprimir_async* crea una orden que contiene el documento que se debe imprimir y lo agrega a la cola de impresión. La función *confirmar* espera a que la orden sea marcada como impresa. La función *terminar_impr* se invoca cuando se terminó la impresión de todos los documentos. En *terminar_impr* Ud. debe enterrar adecuadamente todos los threads consumidores (con *pthread_join*).

Su solución no debe tener *dataraces* como tampoco ciclos de *busy-waiting*. Para la sincronización use un mutex y una o más condiciones.

Este problema es similar al problema del productor/consumidor. Los productores son los threads que imprimen los documentos y los consumidores son los threads que Ud. crea para imprimir los documentos. La diferencia es que en este problema un productor puede esperar hasta que su ítem sea consumido llamando a la función *confirmar*. Otra diferencia es que la cola de impresión (el *buffer*) es de tamaño infinito.

Entregue su tarea solo si el programa de prueba lo felicita señalándole que funciona correctamente. Además se inspeccionará el código de su tarea para verificar el uso correcto de los mutex y condiciones, y que los threads creados son enterrados adecuadamente. Compile y pruebe su tarea en *anakena.dcc.uchile.cl*. La compilación de su tarea no debe arrojar warnings.

Recursos

Baje el archivo *t5.zip* y descomprímalo. Ahí encontrará el *Makefile* para compilar su tarea, el archivo *impr.h* con los encabezados de las funciones pedidas y dadas, y el archivo *test-impr.c* que contiene la función *main* con el test de prueba, la función *imprimir* y una implementación simple de una cola fifo.

```
int main();
void imprimir(Doc *doc, Impr *impr);
```

Este problema es similar a la pregunta 2 del [control 2 del semestre otoño de 2014](#). Lea el enunciado porque le será de ayuda para resolver el problema. Lo único que tendrá que pensar es como hacer que termine *hilo_impresor* para poder enterrar los threads que Ud. creo en *init_impr*.

Entrega

Ud. debe entregar el archivo *impr.c* mediante U-cursos. No incluya otros archivos por favor. No se aceptarán tareas que no pasen el test de prueba que se encuentra en *test-impr.c*. Se descontará medio punto por día de atraso (excepto sábados, domingos y festivos). Resuelva esta tarea antes del control 3. Le servirá de estudio.