

# Trabajo teórico-práctico voluntario. Ejercicio de Monitores POSIX con C++ (librería Pthreads)

Curso 2022-23

## Resumen

Este documento contiene la especificación, objetivos, materialización y evaluación del trabajo teórico/práctico a realizar por el estudiante y que representa una actividad **NO obligatoria y recuperable** según la guía docente de la asignatura Programación Concurrente y Tiempo Real. Esta actividad, cuya temática es la programación concurrente utilizando librerías y lenguajes de alto nivel (concretamente, monitores POSIX implementados con la librería Pthreads y lenguaje C++), tiene un peso del 15% de la nota final y se divide en dos partes:

1. La implementación de un ejercicio/proyecto relativo a la utilización de lenguajes de alto nivel en el contexto de la Programación Concurrente y el Tiempo Real (en lugar de utilizar ANSI C y las primitivas básicas de POSIX vistas en la asignatura al trabajar con *semáforos* y *colas de mensajes*). Concretamente, se plantea resolver el problema clásico del **Puente de un solo carril**, en su variante más sencilla, empleando lenguaje C++ y la librería Pthreads, que cumple el estándar POSIX. **Esta parte se puede realizar de manera individual o por parejas. Si se realiza por parejas, los dos miembros han de realizar la entrega del documento** (ver Sección de “Entrega y normas de presentación”).
2. La realización de un **test individual** que está previsto abordarse, inicialmente, el mismo día de la convocatoria ordinaria (o extraordinaria en su caso). Se trata de un test corto de aproximadamente 10-12 min de duración sobre **conceptos teóricos y prácticos** adquiridos al implementar el ejercicio aquí propuesto (así como sobre **otros conceptos que se vean en las clases de teoría del TEMA 4** del libro de referencia y TEMA 4 en Campus Virtual: “*Otros Mecanismos de Comunicación y Sincronización*”). **Para evaluar el trabajo teórico-práctico es necesario tanto haber entregado el ejercicio (parte 1) como la realización del test (parte 2).**

## Objetivo

- Refuerzo de los conceptos teóricos y las destrezas de programación concurrente a la hora de utilizar los mecanismos de comunicación y sincronización propuestos, dotados de un nivel de abstracción mayor que los clásicos (semáforos y colas de mensajes).

## Especificación

El ejercicio/proyecto a realizar es expuesto a continuación.

## Ejercicio relativo a la implementación de Monitores POSIX con C++ y librería Pthreads

Implemente, mediante **monitores** POSIX, utilizando el lenguaje C++ y la librería Pthreads, un sistema que simule el problema del “Puente de un solo carril” visto en varias ocasiones en clase.

Para facilitar las cosas, se **proporciona un archivo .zip con una plantilla de código fuente** con partes **semicompletas** y un archivo **makefile**. En el propio código se indica **mediante comentarios** algunos consejos y pistas sobre lo que falta por completar en los archivos .cpp correspondientes de la citada plantilla.

\*. Para resolver el ejercicio, se recomienda **analizar con detalle el problema del Buffer Limitado resuelto con monitores** que comentaremos exhaustivamente en clases de teoría.

### Entrega y normas de presentación

Cada estudiante deberá subir un único archivo comprimido cuyo nombre consistirá en *<apellido\_nombre>.zip* (ejemplo: *garcia\_andres.zip*). Si el trabajo se realizar por parejas, **cada uno de los integrantes de las mismas ha de realizar la entrega**,

Al descomprimir el fichero, se deberá obtener la siguiente estructura de directorios/archivos:

<b>memoria.pdf</b>	Contendrá una pequeña memoria explicativa en formato pdf.
<b>puente_un_carril</b>	Directorio que contendrá el código fuente compilable y completo (implementando todo lo necesario, esto archivos .cpp y .h), así como el <b>makefile</b> .

La **memoria** del trabajo consiste en un documento .pdf con una breve discusión (no más de una 1 página) sobre la solución alcanzada para el ejercicio/proyecto **explicando, únicamente, lo que se ha desarrollado en el monitor** (Puente .cpp) y como las operaciones de éste son invocadas desde objetos de clase Coche (*Coche.cpp*). También, se debe añadir en la memoria **un ejemplo de ejecución (captura de la salida por terminal)**.

Para las memorias entregadas, que corresponden con trabajos realizados en pareja, se debe **indicar en la portada el nombre de los dos integrantes**.

El trabajo/proyecto optativo deberá entregarse como máximo **el día 30 de mayo de 2022 a las 23:55 horas** a través de la tarea correspondiente que se habilitará en el espacio virtual de la asignatura. **No se tendrán en cuenta las memorias entregadas fuera de plazo. No se admitirán memorias entregadas por e-mail.** La entrega de la memoria tendrá que realizarse exclusivamente en formato electrónico.

### Evaluación

- El **50% de la nota** del trabajo teórico-práctico voluntario será destinado a la **parte 1ª** (implementación del ejercicio/proyecto en C++ y Pthreads y su memoria).
- El **50% restante de la nota** se destinará a evaluar el grado de adquisición de competencias sobre lo visto en clases de teoría acerca de “*Otros Mecanismos de Comunicación y Sincronización*”, con la realización del test individual (**parte 2ª**).

### Bibliografía básica

El propio libro de referencia de la asignatura.



Facultad de Ciencias Sociales de Talavera de la Reina  
Universidad de Castilla-La Mancha

## **Programación Concurrente y de Tiempo Real**

[BURNS03] Burns, A., Wellings, A., Sistemas en Tiempo Real y Lenguajes de Programación (3ª Edición), Addison-Wesley 2003.

### **Anexo. Plantilla código C++ (partes por completar) y *makefile***

Consultar zip adjunto.