

# **Universidad Católica de Córdoba**

## **Facultad de Ingeniería**

### **Sistemas de Tiempo Real - Año 2008**

**Titular: Ing. John Coppens**

**Prácticos: Ing. Alejandro Butti**

#### **1 Objetivos generales**

Comunicar a los estudiantes el concepto de ejecución paralela de tareas, la comunicación entre ellas, y la sincronización de las mismas. Aplicar los conocimientos en aplicaciones prácticas y lograr la aceptación de los paradigmas de multiprocesamiento y procesamiento en tiempo real.

#### **2 Objetivos específicos**

En la actualidad, aunque se utiliza multiprocesamiento y paralelismo comunmente, y aunque se considera el poder de procesamiento de máquinas modernas 'tiempo real', es necesario estudiar detenidamente las consecuencias y exigencias que tienen estos términos en la vida real. Además - regularmente será necesaria la capacidad de seleccionar el medio de computación apropiado a la tarea.

La materia muestra los efectos de estas técnicas, mediante conceptos teóricos, aplicaciones de alto nivel, y, en forma práctica, con módulos didácticos representando procesos industriales en forma realistica.

#### **3 Síntesis conceptual de la asignatura (Abstract)**

##### **3.1 Unidad 1 y 2**

Introducción: Terminología Términos más frecuentes, definición de conceptos de concurrencia. Ejemplos sencillos en C, para mostrar las clases de concurrencia. Un ejemplo en assembler para microcontrolador de un microsistema concurrente.

##### **3.2 Unidad 3**

Controladores PID Introducción al control de procesos Algoritmo Proporcional/Integral/Diferencial. La necesidad física de una controlador que contenga los tres elementos.

##### **3.3 Unidad 4, 5 y 6**

Pascal FC - Procesos Conceptos de procesos - secuenciales y concurrentes. Representación matemática. Condiciones especiales y consecuencias. Estudio detallado de las herramientas clásicas de exclusión mutua y sincronización. Implementación en Pascal-FC, discusión y resolución de problemas.

##### **3.4 Unidad 6**

Algunos ejemplos de uso en sistemas operativos comunes, y algunos ejemplos de sistemas operativos específicamente desarrollados para concurrencia/tiempo real. Ejemplo de uso de PLC en la implementación de control industrial.

## 4 Contenidos

### 4.1 Unidad 1

Terminología Unidad - Terminología - clasificación de programas por funcionamiento: cooperativos - preemptivos, por respuesta: sincrónicos - asincrónicos, por punto de implementación: hardware - firmware - sistema operativo - compilador - programa usuario.

### 4.2 Unidad 2

La vida real Ejemplos sencillos de diferentes formas de concurrencia en C. Un ejemplo de un sistema concurrente sencillo en assembler para microcontroladores. Ejemplo de un sistema operativo preemptivo implementado en C.

### 4.3 Unidad 3

Controladores PID Descripción y abstracción de un proceso industrial (histeresis, inercia, función de transferencia, estabilidad y otras características de procesos) - realimentación proporcional - corrección de los errores mediante función integral - aumento de velocidad por función diferencial - motor simulado - análisis del sistema completo realimentado. ¿Cual es el efecto de cada uno? ¿Y cuales son los problemas que producen? Criterio de estabilidad del lazo.

Pascal FC Introducción de Pascal-FC como herramienta didáctica. Descripción del compilador, sintaxis, uso, limitaciones. Uso de la versión original DOS, y de la versión adaptada en la Universidad para uso bajo Windows, y otra versión para Linux. Extensiones al idioma realizados para el control de los módulos didácticos.

### 4.4 Unidad 4

Procesos de tiempo real El concepto de proceso - estructuras de procesos - representación en Pascal-FC - representación simbólica - consecuencias de concurrencia.

### 4.5 Unidad 5

Exclusion mutua y sincronización La necesidad de comunicación entre procesos - y los problemas. Mensajería sincrónica. Semáforos. Regiones críticas y monitores.

### 4.6 Unidad 6

Otra vez, la vida real Comparación de la implementación de exclusión mutua y sincronización en sistemas de la vida real: sistemas operativos e idiomas. Herramientas comerciales disponibles, comparacion. Como se implementa control de procesos en la industria: introducción a los PLC.

Introducción al control de procesos Algoritmo Proporcional/Integral/Diferencial. La necesidad física de un controlador que contenga los tres elementos.

Conceptos de procesos - secuenciales y concurrentes. Representación matemática. Condiciones especiales y consecuencias.

- representación en Pascal-FC - representación simbólica - consecuencias de concurrencia.

## 5 Bibliografía

### 5.1 Bibliografía obligatoria

- *Concurrent Programming* Allan Burns, Geoff Davies, Addison Wesley, ISBN 0-201-54417-2
- *Modern Operating Systems* Andrew S. Tanenbaum, Prentice Hall, ISBN 0-13-588187-0

- Material didáctico para la materia  
John Coppens, Sitio material didáctico <http://www.jcoppens.com/univ/treal>

## 5.2 Bibliografía de consulta

- *Algorithms + Data Structures = Programs*  
Niklaus Wirth, Prentice Hall, Inc., ISBN 0-13-022418-9
- *Programming and Problem Solving with Pascal*  
Michael Schneider, Steven Weingart, David Perlman, John Wiley, ISBN 0-471-08216-3
- *Advanced Programming and Problem Solving with Pascal*  
Michael Schneider, Steven Bruell, John Wiley, ISBN 0-471-01128-2
- *Advanced Programming in the UNIX Environment*  
Richard Stevens, Addison Wesley, ISBN 0-201-56317-7
- *Writing Windows Virtual Device Drivers*  
David Thielen, Bryan Woodruff, Addison Wesley, ISBN 0-201-48921-X
- *Beginning Linux Programming*  
Neil Matthew, Rick Stones, Wrox Press Ltd, ISBN 1-874416-68-0
- *Linux Device Drivers, 2nd edition*  
Alessandro Rubini, Jonathan Corbet, O'Reilly, ISBN 0-596-00008-1

## 6 Metodología

El aspecto teórico será presentado con exposiciones didácticas, con énfasis sobre la implementación de ejemplos. La mayoría de los ejemplos se realizarán mediante Pascal-FC, pero casos donde sea más apropiado, se utilizarán otros medios.

En particular, la parte práctica del curso se dedicará a implementar trabajos bajo condiciones reales:

1. Estudio de un sistema operativo didáctico en C
2. Implementación de un protocolo de comunicación
3. Estudio e implementación de algoritmos PID
4. Control de módulos didácticos

## 7 Calendario

### 7.1 Teórico

Fecha	Clase	Unidad	Tema/actividad
11/03	1		Introducción a la materia - Organización del curso - Documentación, prácticos, exigencias - Medios de comunicación (e-mail, lista de discusión, página web)
18/03	2	1	Terminología - Conceptos básicos de qué son los sistemas de tiempo real.
25/03	3	2	La vida real - Un ejemplo de extrema sencillez de un sistema de tiempo real, implementado en assembler. Elementos básicos - elemento tiempo, sincronización.
01/04	4	2	La vida real - Más complicado: sistemas cooperativos/preemptivos. Comparación de sistemas operativos de tiempo real comerciales.
08/04	5	3	Controladores PID - el elemento de control básico para controlar procesos industriales. Como se comportan procesos industriales. Significado físico de los parámetros, y como eliminarlos. Problemas del muestreo. Estabilidad de un lazo de control.
15/04	6	4	Pascal FC. Descripción del idioma, y diferentes implementaciones, para DOS, Windows, y Linux. Extensiones agregados para controlar los módulos didácticos.
22/04	7	5	Procesos: el concepto de ejecución en paralelo, de la independencia entre procesos.
29/04	8		<i>Primer parcial</i>
06/05	9	6	Exclusión mutua y sincronización, Elementos originales: Variables globales para resolver problemas de zonas críticas. Semáforos binarios y genericos. Operaciones permitidas. Implementación de buffers.
13/05	10	6	Exclusión mutua y sincronización: Mas avanzados: Comunicación entre procesos para implementar sincronización y exclusión mutua, via uso de canales.
20/05	11	6	Exclusión mutua y sincronización: Invocacion remota, monitores
27/05	12	6	Exclusión mutua y sincronización: Unificado con recursos.
03/06	13	6	Otra vez, la vida real: Como se implementan estos sistemas en herramientas comerciales? Cuales compiladores están disponibles, comparación.
10/06	14	7	<i>Segundo parcial</i>
17/06	15	7	Revisación de la materia (y recuperatorio)

## 7.2 Prácticos

Fecha	Clase	Unidad	Descripción
14/03	1		Presentación Personal y de los objetivos y alcances de la parte practica en la materia. Introducción al trabajo práctico a desarrollar durante el año
21/03			Viernes Santo
28/03	2	1	Desarrollos de Conceptos básicos sobre el trabajo de Cecilia Tamagusuku para implementar un sistema concurrente
04/04	3	2	Finalizar la explicación del desarrollo del SO de Cecilia Tamagusuku. Solicitud del 1er trabajo práctico
11/04	4		Revisión sobre el TP 1. Y reafirmación de conceptos en base a ese TP desarrollado
18/04	5	3	Introducción al TP2, refrescando conceptos de algoritmo para PID. Explicación sobre el TP2 a desarrollar.
25/04	6		Revisión sobre el TP2. Revisando conceptos de lazo de control PID.
02/05	7	5	Explicación del protocolo de comunicación con los módulos prácticos. Explicación del comportamiento de cada uno de los módulos. Solicitud del TP3.
09/05	8		Inspección del TP3 verificando posibles problemas con el protocolo de comunicación y verificando conceptos sobre el lazo de control PID
16/05	9		Resumen de los TP realizados hasta el momento reafirmando conceptos básicos. Introducción al último TP. Del año
23/06	10		Clase de consulta sobre el TP4 y revisión de lo realizado hasta el momento
30/05	11		Entrega del TP4 finalizado. Revisión de los conceptos básicos y su implementación.
13/06	12	6	Introducción a los conceptos básicos de PLC. Visualización de videos de la aplicación de estos elementos en la industria.
20/06	15		Ultimo plazo para entrega y revisión del TP4. Definición de la regularidad del alumno

## 8 Trabajos prácticos

Para la parte teórica: resolución de problemas de concurrencia, principalmente utilizando el compilador Pascal-FC, para evaluar la comprensión de la materia y de los conceptos.

Para la parte práctica en sí: estudio de un sistema operativo de tiempo real preemptivo (en su momento desarrollado por alumnas de la Universidad). Adición de tareas simples al sistema operativo. Desarrollo de un controlador para procesos industriales. Énfasis sobre la programación estructurada. Simulación de un motor. Implementación del algoritmo PID. Comunicación con los módulos didácticos. Graficación.

## 9 Criterios y formas de evaluación

### 9.1 Durante el semestre

Tanto en la parte teórica como en la parte práctica, se presentarán 4 trabajos durante el semestre. Estos trabajos se evaluarán por su: originalidad (~30%), solidez de implementación (~30%), estructura (~20%), documentación (~20%).

Se considera importante en la documentación, que conste de:

- Descripción del problema. ('La tarea')
- Análisis del problema. ('Como se sugiere solucionarlo')

- Descripción de la solución. En la descripción no se trata necesariamente de un reporte verboso, sino de una representación sintética de la solución adoptada.
- Relato de problemas encontrados, y las soluciones propuestas y/o realizadas.

## **9.2 Parciales**

En la parte teórica se realizarán además dos parciales, evaluando el conocimiento sobre la materia en forma individual. Si se considera conveniente, se reemplazarán el parcial por un trabajo práctico, si esto es mas conveniente para permitir mayor creatividad y aplicación de los conocimientos. En tal caso, se hará una evaluación personalizada de los resultado.

En las notas finales se agregará una evaluación de la participación del alumno en las actividades.

## **9.3 Final**

El examen final teórico consistirá en dos partes: una evlucción de los conocimientos teóricos de la materia y una confirmación individual de la participación en la realización de los trabajos presentados.

El responsable de la parte práctica, se encargará de la misma manera de evaluar los conocimientos absolutos e individuales de los alumnos.

## **10 Condiciones para obtener la regularidad**

Las condiciones formales son las que actualmente rigen para todas las cátedras, según el reglamento vigente.