



# Planificaciones

7559 - Técnicas de Programac. Concurrente I

Docente responsable: GARIBALDI JULIA ELENA

## OBJETIVOS

Familiarizar al estudiante con las técnicas de Programación Concurrente, los conceptos y los algoritmos correspondientes al manejo de procesos cooperantes, con ambiente propio (threads) o sin él, la corrección de programas concurrentes. Introducir los principios de diseños de algoritmos cooperativos usando un lenguaje de programación y los recursos de sincronización como mensajes, semáforos, monitores, regiones críticas condicionales y barreras. Introducir al estudio de las Redes de Petri. Introducción a los problemas de concurrencia de transacciones distribuidas. Profundizar el conocimiento de subsistema de IPC en UNIX- Linux.

## CONTENIDOS MÍNIMOS

-

### PROGRAMA SINTÉTICO

Principios de la programación concurrente. Sección crítica. Exclusión Mutua. Corrección de programas: propiedades safety y liveness. Sincronización y comunicación. Semáforos, Monitores. Problemas clásicos y su solución con diversos recursos de sincronización y comunicación. Paso de mensajes. Redes de Petri. Transacciones distribuidas.

### PROGRAMA ANALÍTICO

1. Unidad 1: Conceptos básicos de la programación concurrente.

1.1. Introducción

1.2. Programación secuencial vs programación concurrente.

1.3. Necesidad de la programación concurrente.

1.4. Problemas en la programación concurrente.

1.5. Arquitecturas de hardware que soportan la programación concurrente.

2. Unidad 2: Representación y ciclo de vida de los procesos.

2.1. Concepto de proceso.

2.2. Estructura de un proceso.

2.3. Ciclo de vida de un proceso.

2.4. Planificación de procesos.

3. Unidad 3: Interacción de procesos.

3.1. Entidades pasivas y activas.

3.2. Exclusión mutua y otros métodos.

3.3. Corrección de programas concurrentes. Propiedades de tipo "safety" y "liveness". Lógica temporal. Propiedades e invariantes.

3.4. Sincronización con variables compartidas.

3.5. Primitivas de sincronización para variables compartidas.

3.6. Modelos de problemas clásicos y soluciones con variables compartidas, espera ocupada.

4. Unidad 4: Semáforos

4.1. Introducción

4.2. Tipos de semáforos

4.3. Invariantes de semáforo.

4.4. Problema del productor-consumidor

4.5. Buffers.

4.6. Problema de lectores- escritores.

4.7. Problema de filósofos comensales.

5. Unidad 5: De regiones críticas a monitores

5.1. Introducción

5.2. Regiones críticas como modelo de solución de exclusión mutua

5.3. Regiones críticas condicionales

5.4. Monitores

5.5. Comparación entre los diferentes modelos.

6. Unidad 6 : Sistemas basados en paso de mensajes

6.1. Introducción a los mecanismos de paso de mensajes en los lenguajes de programación concurrente.

6.2. El modelo de programación distribuida CSP. Sintaxis y semántica de las órdenes de la notación CSP.

6.3. Mecanismos de alto nivel en sistemas y lenguajes de programación.

6.4. Modelos de comunicación asíncrona vía mensajes

- 6.5. Modelos de comunicación sincrónica via mensajes
- 6.6. Paradigma cliente-servidor.
- 6.7. Idempotencia.

## 7. Unidad 7: Modelado de sistemas concurrentes.

- 7.1. Redes de Petri Lugar/Transición (LT)
- 7.2. Clasificación de redes de Petri
- 7.3. Métodos de análisis de propiedades de redes LT. Árbol de alcance. Invariantes de marcado y disparo.
- 7.4. Métodos de análisis basados en el Álgebra Lineal de Matrices.
- 7.5. Aplicaciones.

## 8. Unidad 8: Introducción a la concurrencia en ambientes distribuidos.

- 8.1. Introducción.
- 8.2. Exclusión mutua distribuida.
- 8.3. Algoritmos de exclusión y elección.
- 8.4. Transacciones
  - 8.4.1. Transacciones anidadas.
  - 8.4.2. Bloqueos.
  - 8.4.3. Métodos de control de concurrencia
- 8.5. Transacciones distribuidas
  - 8.5.1. Transacciones planas y anidadas.
  - 8.5.2. Control de la concurrencia.
  - 8.5.3. Interbloqueos distribuidos.
  - 8.5.4. Recuperación de transacciones.

## BIBLIOGRAFÍA

Nota: para cada libro indicado, utilizar la última edición disponible.

- Taubenfeld, Gadi, Synchronization algorithms and concurrent programming, Pearson/Prentice Hall, 2006
- Ben-Ari, "Principles of Concurrent Programming".
- Stallings, W., "Sistemas Operativos".
- Tanenbaum, A., "Sistemas Operativos Modernos".
- Burns, Alan and Wellings Andy, "Concurrente and Real-Time Programming in Ada", Cambridge University Press, 2007.
- Burns, Alan, "Concurrent Programming".
- Bach, "El diseño del sistema operativo Unix".
- Lea, D., "Programación concurrente en Java: principios de diseño y patrones", Addison – Wesley, 1999.
- Coulouris, Dolklilmore, Kindberg, "Sistemas distribuidos, conceptos y diseño", Addison Wesley, cuarta edición, 2005.
- Andrews, Gregory R., Foundations of multithreaded, parallel and distributed programming, Addison-Wesley, 2000.
- Magee, Jeff, Kramer, Jeff, concurrency: state models & Java programs, Wiley, 2006.
- Rochkind, M., "Advanced UNIX programming", Addison Wesley, 2004.
- Butenhof, D, "Programming with POSIX threads", Addison Wesley, 1997.
- Ben-Ari, M, Mathematical logic for computer science, Springer, 2001.
- Apuntes y "papers" a indicar durante la cursada.
- <http://www.advancedlinuxprogramming.com/docs.sun.com>
- Herlihy, M., Shavit, N., The art of multiprocessor programming, Elsevier, 2008
- Santoro, N., Design and analysis of distributed algorithms, John Wiley & Sons, Inc. 2007

## RÉGIMEN DE CURSADA

### Metodología de enseñanza

- Guía de ejercicios por unidad desarrollada.
- Explicaciones y resolución de problemas en laboratorio.
- Dos trabajos prácticos de aplicación a casos de estudio.
- Estudio de papers relativos a temas del programa.

### Modalidad de Evaluación Parcial

De acuerdo a las disposiciones del C.D.

**CALENDARIO DE CLASES**

Semana	Temas de teoría	Resolución de problemas	Laboratorio	Otro tipo	Fecha entrega Informe TP	Bibliografía básica
<1> 09/03 al 14/03	Unidad 1	Ver laboratorio	Práctica 0			Ver Información general
<2> 16/03 al 21/03	Unidad 1	Ver laboratorio	Práctica 1: procesos			Ver Información general
<3> 23/03 al 28/03	Unidad 2	Ver laboratorio	Práctica 2: señales			Ver Información general
<4> 30/03 al 04/04	Unidad 2	Ver laboratorio	Práctica 3: pipes			Ver Información general
<5> 06/04 al 11/04	Unidad 3	Ver laboratorio	Práctica 4: locks		Entrega de ejercicios seleccionados de las guías	Ver Información general
<6> 13/04 al 18/04	Unidad 3	Ver laboratorio	Definición y planteo del primer trabajo práctico			Ver Información general
<7> 20/04 al 25/04	Unidad 4	Ver laboratorio	Práctica 5: semáforos			Ver Información general
<8> 27/04 al 02/05	Unidad 4	Ver laboratorio	Parcial primera fecha			Ver Información general
<9> 04/05 al 09/05	Unidad 5	Ver laboratorio	Práctica 6: Hilos en Java			Ver Información general
<10> 11/05 al 16/05	Unidad 5	Ver laboratorio	Entrega trabajo práctico 1		Entrega trabajo práctico 1	Ver Información general
<11> 18/05 al 23/05	Unidad 6	Ver laboratorio	Práctica 7: Mensajes			Ver Información general
<12> 25/05 al 30/05	Unidad 6	Ver laboratorio	Práctica 8: sockets			Ver Información general
<13> 01/06 al 06/06	Unidad 7	Ver laboratorio	Parcial segunda fecha			Ver Información general
<14> 08/06 al 13/06	Unidad 7	Ver laboratorio	Práctica 9: redes de Petri			Ver Información general
<15> 15/06 al 20/06	Unidad 8	Ver laboratorio	Práctica 10: transacciones distribuidas			Ver Información general
<16> 22/06 al 27/06	Unidad 8	Ver laboratorio	Entrega trabajo práctico 2		Entrega trabajo práctico 2	Ver Información general

## CALENDARIO DE EVALUACIONES

### Evaluación Parcial

Oportunidad	Semana	Fecha	Hora	Aula
1º	10			
2º	15			
3º				
4º				
Observaciones sobre el Temario de la Evaluación Parcial				
El conjunto de temas a evaluar durante el primer parcial corresponderá a todo lo presentado en clases teóricas o prácticas y también se basará en las guías de trabajos prácticos resueltas hasta una semana antes de la evaluación. Las fechas de evaluación son tentativas. Al inicio del cuatrimestre serán revisadas y comprometidas.				
Otras observaciones				
La tercera oportunidad se fijará a partir de la semana 17 (desde el inicio de la cursada).				