



**UNIVERSIDAD SIMÓN BOLÍVAR**  
**Vicerrectorado Académico**

1. Departamento: *COMPUTACIÓN Y TECNOLOGÍA DE LA INFORMACIÓN*

**2. Asignatura: SISTEMAS DE operación II**

3. Código de la asignatura: CI-4821

No. de unidades-crédito: 3

No. de horas semanales: Teoría 3    Práctica    Laboratorio 2

4. Fecha de entrada en vigencia de este programa: ENERO 95

**5. OBJETIVOS GENERALES:**

- .- Estudiar las diferentes técnicas de especificación de concurrencia intraproceso e interproceso.
- .- Conocer la implementación de las técnicas en lenguajes de programación concurrente como C Concurrente, ADA y/o Occan.
- .- Estudiar las técnicas comunmente utilizadas para diseñar sistemas de operación.
- .- Analizar la implementación de sistemas de operación bajo las técnicas de diseño estudiantes. Entre los sistemas de operación a estudiar se encuentran Minix, Xinu, Linux, Tunis, Amoeba, Mach, Chorus.

**6. CONTENIDO DETALLADO TEORÍA:**

- 1.- Introducción a los procesos concurrentes. Necesidad de la sincronización y la comunicación.
- 2.- Especificación de Concurrencia Intraproceso:

.- Condiciones de Bernstein

- .- Sistemas de Paralelismo Máximo
- .- Especificación basada en Cobegin/Coend
- .- Especificación basada en Fork/Join
- .- Equivalencia entre especificaciones

3.- Especificación de concurrencia interproceso: el problema de región crítica: Descripción y condiciones necesarias y suficientes para preservación de región crítica.

- .- Soluciones para dos procesos
- .- Soluciones para N ( $N > 2$ ) procesos

4.- Problemas clásicos de concurrencia:

- .- Problemas de Lectores y Escritores
- .- Problema de la cena de los filósofos

5.- Primitivas de Sincronizaciones de bajo nivel:

- .- Semáforos. Topificación en base a lo justicio (semaforos justos o no justos)
- .- Secuenciadores y Contadores de Eventos
- .- Estudio de equivalencia de primitivas

6.- Primitivas de sincronización de alto nivel:

- .- Regiones críticas
- .- Regiones críticas condicionales
- .- Monitores
- .- Path Espressions
- .- Estudio de equivalencia de primitivas

7.- Comunicación entre procesos: requerimientos y caracterización

8.- Lenguajes Concurrentes:

- .- ADA
- .- C Concurrentes

9.- Técnicas Diseño de Sistemas de Operación:

- .- Estructuras monolíticas
- .- Estructuras jerárquicas
- .- Microkernels
- .- Upcalls
- .- Diseño cliente/servidor
- .- Diseño orientado por objeto

10.- Análisis de los sistemas de operación seleccionados como caso de estudios

## **7. CONTENIDO DETALLADO PRÁCTICA Y/O LABORATORIO:**

El laboratorio toma un caso de estudio. Para este caso de estudio se evalúa la implementación desde tres perspectivas: Manejador de Memoria, Manejador de Procesos y Manejador de Archivos. El estudiante analizará el código Fuente del Sistema de Operación para determinar la forma de implementación y las técnicas seguidas en las mismas.

Adicionalmente se enfrenta al estudiante con desarrollos de sistemas concurrentes utilizando las facilidades del IPCS (Interprocess Communication System) ofrecidas por el sistema de operación a utilizar. En general, se desarrollan proyectos sobre las siguientes primitivas: Memoria Compartida, Semáforos y Colas de Mensaje, a través de llamadas al Sistema de Operación (System Calls).

Sujeta a la disponibilidad de recursos, el estudiante podrá trabajar con un lenguaje de programación concurrente. Tradicionalmente se utiliza el lenguaje C Concurrente.

## **8. BIBLIOGRAFÍA:**

- .- Tenembaun, A. Operating Systems; Design and Implementation. Prentice Hall, 1987
- .- Bacon, J. Concurrent Systems: An integrated approach to Operating Systems, Databases and Distributed Systems. Addison-Wesley 1992
- .- Material hemerográfico sobre casos de estudio
- .- Tanembaun, A. Distributed Operating Systems. Prentice Hall, 1995 (material para casos de estudio)