

ANOTACIONES:

Definición:

ocurrencia. Ejemplo: **UNIVERSIDAD DE BUENOS AIRES**

concurrentes.

Algoritmo: **FACULTAD DE CIENCIAS EXACTAS Y NATURALES**

desconectando.

DEPARTAMENTO: Computación.....

ASIGNATURA: Programación Concurrente I.....

CARRERA/B: Cs de la Computación (18), (82), (87).....

y Pascal.

*Nociones de Pascal.

CARACTER: optativa..... (indicar si es obligatoria u optativa)

*N/A

PUNTAJE: 3 puntos..... (en caso de ser optativa)

*N/A

DURACION DE LA MATERIA: cuatrimestral..... (indicar si es cuatrimestral o anual).

HORAS DE CLASE: a) TEORICAS...5.. HS. b) PROBLEMAS3.. HS.

c) LABORATORIO... HS. d) SEMINARIOS..... HS.

INTEGRACION: e) TOTALES....8.. HS.

ASIGNATURAS

CORRELATIVAS: Sistemas Operativos. (Trabajos..Prácticos).....

las.....

desarrollar.....

punto 1.

PROGRAMA:

1. Evoluciones primitivas.

Formalismos de representación de procesos.

Procedencias primitivas de implementación: Fork/join y cobegin/

coend; Comparación entre las dos. Sincronización: P/V, await/cause

Regiones críticas condicionales, monitores y clases. Nociones de

implementación.

G. Deibud, 1970.

Ethos, Pandal, 1970.

2. Programación Concurrente.

Programación Concurrente bajo sistemas de multiprogramación.

Reales y/o distribuidos. Concurrent Pascal C, Shell. Concurrent-

tización de programas secuenciales, condiciones de Bernstein.

3. Metodologías para PC.

Formalismo de Redes de Petri, secuencialidad y concurrencia. Aplica-

ciones en el diseño/modelización de grandes sistemas. Editores de

DFD'S. Trade-off concurrencia versus secuencialidad, nociones

intuitivas: overhead y economía de hard. Algunos resultados

básicos.

4. Abrazos mortales.

Definición. Condiciones necesarias y suficientes para su ocurrencia. Ejemplos de la vida real y de programación concurrente. Políticas de prevención, evitando laissezfaire. Algoritmo de asignación ordenada, del banquero y detención y desenredando.

Prerrequisitos:

* Práctica considerable en algún lenguaje de programación entre C y Pascal.

* Nociones de compilación.

* Uso considerable de algún sistema operativo, preferentemente de multiprogramación o 1 año mínimo de experiencia como systems programmer.

* Nociones de Unix y su lenguaje de programación de comandos (Shell).

Objetivos:

Integración del concepto de concurrencia en la programación de procesos complejos y su utilización práctica. Toma de conciencia de las limitaciones y su alcance. Como práctica durante el curso se desarrollará un procesador para implementar construcciones del punto 1, bajo Xenix System V o similar.

Bibliografías:

- P. B. Hansen, "Operating Systems Principles", Prentice-Hall, 1974.
- Gravello (2nd. edition), 1984.
- Manual de uso del Shell, en Unix/Xenix (CP).
- W. Rasegg, "Petri Nets", Springer Verlag, New York, 1985.
- G. Delbue, "Programación Concurrente". II Encuentro Proyecto Ethos, Tandil, 1987.

Fecha 15 de marzo de 1990.

Profesor Responsable

G. Delbue

Intuitivas - overhaul
básicos.

Autoridad Departamental

Lic. ADOLEO EL XVICCA

COMITÉ DE CONSULTACIÓN