



N° DISPOSICIÓN:

Universidad Nacional de Luján
República Argentina

Ruta 5 y Av. Constitución
C.C. 221 - 6700 - LUJÁN (Bs. As.)

DEPARTAMENTO DE: **Ciencias Básicas**

CARRERA/S: **Licenciatura en Sistemas de Información**
(RES.HCS. N°238/04)

PROGRAMA DE LA ASIGNATURA: **Taller Libre II (11441)**

RESPONSABLE: Mg. Gabriel H. Tolosa , Profesor Adjunto	HORAS DE CLASE SEMANALES: 6
EQUIPO DOCENTE: Lic. Santiago Bancho , Ayudante de Primera	TEÓRICAS: 3 PRÁCTICAS: 3
	HS.TOTALES: 96
ASIGNATURAS CORRELATIVAS	
CURSADAS	APROBADAS
11421 (Regular para cursar)	11421 (Para aprobar)
VIGENCIA: 2012	

CONTENIDOS MÍNIMOS: Según RES.HCS. N°238/04

Esta asignatura tiene tema libre. Su objetivo es el de actualizar a los alumnos en temas de reciente desarrollo. Por esta razón, la Coordinación de la carrera definirá para cada año los temas a tratar.

FUNDAMENTACIÓN

Los requerimientos de cómputo de numerosas aplicaciones obligan a desarrollar software eficiente. Sin embargo, algunos problemas han alcanzado los límites físicos de la computación secuencial, haciendo necesaria una solución que utilice múltiples procesadores. Esto requiere de técnicas de programación especiales que permitan manejar la concurrencia en el uso de recursos y/o resolver el problema mediante computación paralela. Las máquinas paralelas permiten disminuir el tiempo total de ejecución de un programa para la solución de un problema, resolviendo problemas de mayor tamaño que en una máquina secuencial no sería posible de abordar.

Además, el auge de procesadores con múltiples núcleos y de las redes de datos han aumentado la difusión del procesamiento concurrente y paralelo. Existen diferentes arquitecturas de máquinas paralelas (por ejemplo, sistemas SMP, clusters) y modelos de programación, con diferentes niveles de abstracción, que se utilizan para la solución de un muy variado número de problemas: desde la ingeniería al cómputo científico, pasando por la física, la bioinformática, la simulación, el rendering de gráficos, el procesamiento de señales, entre otras.

OBJETIVOS GENERALES:

En este taller se propone presentar los fundamentos de la programación paralela y concurrente y las problemáticas asociadas. Se presentan las diferentes arquitecturas de máquinas paralelas y los modelos de programación vigentes.

Además, se plantea una alta carga de prácticas de programación en laboratorio resolviendo problemas de naturaleza diversa. Desde los algoritmos clásicos de tratamiento algebraico paralelo (principalmente de matrices) hasta algunos problemas de la física. Se abordarán modelos de programación basados tanto en pasaje de mensajes como en aquellos para sistemas con memoria compartida.

Se requiere que los alumnos cuenten con una base sólida en programación y conocimientos de redes ya que las actividades prácticas requerirán de estas habilidades. Sobre el cierre del taller se proponen temas para que los alumnos realicen un proyecto final en el cual deberán resolver un problema concreto mediante un enfoque paralelo.

A continuación se presentan los principales objetivos de este curso, basados en los conocimientos que se propone adquirir y las habilidades a desarrollar. Concretamente, se espera que los alumnos:

- Comprendan la problemática de la programación paralela tanto desde la arquitectura como de los lenguajes.
- Conozcan los modelos de computación paralela y los paradigmas de programación paralela.
- Conozcan las métricas de performance necesarias para evaluar un sistema paralelo.

- Sean capaces de abordar la transformación de un algoritmo secuencial a paralelo.
- Conozcan la problemática de los sistemas concurrentes y los algoritmos fundamentales que permiten la solución de problemas de concurrencia.
- Conozcan los fundamentos de comunicación y sincronización entre procesos, tanto para sistemas con memoria compartida como de pasaje de mensajes.

Complementariamente, se propone que aquellos que participen de la asignatura también incrementen sus habilidades para:

- Redactar informes de desarrollo, reportes técnicos o trabajos de investigación siguiendo objetivos y metodología concreta.
- Comunicar sus conocimientos, resultados de investigación a pares y/o superiores en presentaciones públicas.

CONTENIDOS:

Unidad 1: Conceptos sobre Sistemas Distribuidos

Introducción a los sistemas distribuidos. Caracterización. Paradigmas de computación distribuida. Modelos de sistemas y soporte de comunicaciones. Comunicación entre procesos: RPC, API de sockets.

Unidad 2: Gestión de Recursos Distribuidos

Sistemas de Archivos Distribuidos. Arquitectura. Casos de estudio: NFS. Servicios de Nombres. Servicios de directorio y descubrimiento.

Unidad 3: Tiempo y Estados Globales

El problema de la sincronización global. Relojes, eventos y estados. Sincronización. Relojes lógicos. Estados globales. Algoritmos clásicos: Lamport.

Unidad 4: Sistemas Peer-to-Peer

Modelo P2P vs Cliente/Servidor. Características de los nodos. Sistemas puros e híbridos. Redes a nivel de aplicación. Protocolos clásicos: Gnutella, Freenet, DHTs. Eficiencia. Aplicaciones clásicas.

Unidad 5: Introducción a la programación concurrente

Objetivos de los sistemas concurrentes. Programa concurrente. Clases de aplicaciones. Multithreading. Conceptos sobre algoritmos concurrentes, distribuidos y paralelos. Requerimientos. Sincronización y comunicación: Sincronización por exclusión mutua y por condición. Comunicación por memoria compartida y por mensajes.

Unidad 6: Concurrencia y sincronización

Aspectos de programación secuencial. Especificación y semántica de la ejecución concurrente. Acciones atómicas y sincronización. Requerimientos para los lenguajes de programación. Sincronización por variables compartidas, semáforos y monitores. Programación distribuida. Concurrencia con pasaje de mensajes. Mensajes asíncronos y sincrónicos. RPC y Rendezvous.

Unidad 7: Introducción a la programación paralela

Objetivos del procesamiento paralelo. Arquitecturas y clasificación. Interacción, comunicación y sincronización de procesos. Concepto de Sistema Paralelo. Eficiencia básica de los algoritmos paralelos. Medidas de performance standard. Speedup y eficiencia. Modelo de carga fija (Amdahl). Modelo de tiempo fijo (Gustafson). Escalabilidad.

Unidad 8: Programación paralela con pasaje de mensajes

Principios de la comunicación/sincronización por pasaje de mensajes. Primitivas Send y Receive. La interfaz MPI como

modelo. Cómputo y Comunicaciones. Comunicaciones colectivas y operaciones de procesamiento.

Unidad 9: Programación paralela con memoria compartida

Concepto de thread. Ventajas y desventajas con los procesos. Primitivas de sincronización y control de atributos en threads. Modelo OpenMP. Análisis de problemas.

Unidad 10: Algoritmos paralelos clásicos

Ordenamiento. Algoritmos sobre grafos. Tratamiento numérico de ecuaciones diferenciales y de álgebra lineal (Factorización LU, Matrices ralas, métodos iterativos, producto de matrices). Generación de números aleatorios.

METODOLOGÍA:

El desarrollo del curso es de carácter teórico - práctico, con alto requerimiento de participación por parte de los alumnos. En las clases teóricas se plantearán los fundamentos, conceptos, modelos, ejemplos y aplicaciones de los temas cubiertos por el programa.

En las clases prácticas se realizarán ejercicios de programación concurrente y paralela, fundamentalmente para entender los problemas a enfrentar. Se trabajará sobre el sistema operativo Linux y - en general - sobre el lenguaje de programación C o C++.

Complementariamente, los alumnos deberán realizar de manera grupal la lectura e investigación sobre un tema para preparar una exposición. Se espera que en ésta se pueda discutir en clase con todo el grupo sobre las características y aspectos relevantes del tema asignado.

Finalmente, se propondrá la realización de un trabajo final de curso consistente en la solución en paralelo de un problema dado. Los requerimientos funcionales de la aplicación podrán ser propuestos por los alumnos y aprobados por el docente.

ACTIVIDADES PRÁCTICAS

Por tratarse de una asignatura con modalidad taller, las actividades prácticas son puramente de laboratorio y permiten reforzar los conceptos planteados en clase a partir de su implementación en software. Aquí se deberán realizar programas orientados a diferentes problemas.

Las aplicaciones podrán estar escritas en lenguaje C o C++ y los alumnos deberán demostrar sus habilidades en la programación como así también en el análisis de las opciones (si hubiese) para la construcción de la solución. Siempre contarán con el soporte del equipo docente.

Para el trabajo final, los alumnos presentarán su propio proyecto, el cual discutirán con los docentes. Además, deberán presentar un informe (en un formato que se especificará de acuerdo a la naturaleza del proyecto) donde se exponga la problemática a resolver, la propuesta, la metodología utilizada y los resultados obtenidos. Opcionalmente, deberán exponer su trabajo en clase.

EVALUACIÓN

La evaluación consta de un examen parcial y un trabajo final integrador (descrito en el apartado anterior) obligatorio. El examen parcial se aprueba con nota 4 (cuatro) o superior mientras que el integrador con 7 (siete) o superior.

Al finalizar, existe una instancia de recuperatorio para quien no haya aprobado el examen parcial. Las condiciones luego de cursar la asignatura son las siguientes:

PROMOVIDO:

- Aprobar todos los trabajos prácticos y/o actividades académicas especiales previstas.
- Aprobar el cien por ciento (100%) de las evaluaciones previstas (2 parciales) con un promedio final no inferior a seis (6) puntos, sin haber recuperado ninguna.
- Aprobar la evaluación integradora (TP Final Integrador) de la asignatura con calificación no inferior a siete (7) puntos.

REGULAR:

- Aprobar todos los trabajos prácticos y/o actividades académicas especiales previstas.
- Aprobar el cien por ciento (100%) de las evaluaciones previstas (2 parciales) con una calificación no inferior a cuatro (4) puntos. Se podrá recuperar una de las instancias.
- Aprobar la evaluación integradora (TP Final Integrador) de la asignatura con calificación no inferior a cuatro (4) puntos.

LIBRE:

- Es aquel que habiendo participado en al menos una (1) de las evaluaciones establecidas como obligatorias en el programa oficial de la asignatura, o de las instancias de recuperación de la misma, no hubiera alcanzado el rendimiento exigido para ser considerado regular.
- Para el examen libre: Quince días antes de la fecha de sustanciación de mesa, el alumno deberá entregar la resolución de todas las actividades prácticas vigentes en la última cursada.

AUSENTE:

- Es aquél que habiéndose inscripto en la asignatura, no ha cumplido con ninguna de las actividades evaluables establecidas por el programa oficial de la misma.

Las condiciones de asistencia se rigen según el Capítulo II y V del Régimen General de Estudios de la Universidad.

BIBLIOGRAFÍA

SUGERIDA

- Colouris, G., Dollimore, J., Kindberg, T. Sistemas Distribuidos. Conceptos y Diseño. 4ta ed. Addison Wesley, 2006.
- Liu, M. Computación Distribuida. Fundamentos y Aplicaciones. Addison Wesley, 2004.
- Pacheco, P. Parallel Programming with MPI. Morgan Kaufmann; 1st edition, 1996.
- Ben-Ari, M. Principles of Concurrent and Distributed Programming, 2/E. Addison-Wesley. 2006.
- Gama A., Gupta A., Karypis G., Kumar V., An Introduction to Parallel Computing. Design and Analysis of Algorithms, Pearson Addison Wesley, 2nd Edition, 2003.

DE CONSULTA

- Pacheco, P. An Introduction to Parallel Programming. Morgan Kaufmann; 1 edition, 2011.
- Chandy, Misra, "Parallel Program Design. A Foundation", Addison Wesley, 1988.
- Jordan H.F., Alaghband G., Jordan H.E., "Fundamentals of Parallel Computing", Prentice Hall, 2002.
- Leopold C., "Parallel and Distributed Computing. A survey of Models, Paradigms, and Approaches", Wiley Series on Parallel and Distributed Computing. Albert Zomaya Series Editor, 2001.

RECURSOS ADICIONALES

El equipo docente mantiene un sitio web de la asignatura (<http://www.tyr.unlu.edu.ar/TallerII/>) con un blog en el cual se publica el material regular y las novedades. Además, se atienden durante todo el año consultas por correo electrónico y/o sesiones de chat.