



Ministerio de Cultura y Educación
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area II: Sistemas de Computacion

(Programa del año 2024)
(Programa en trámite de aprobación)
(Presentado el 16/04/2024 10:04:13)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
SISTEMAS DISTRIBUIDOS Y PARALELISMO	LIC.CS.COMP.	32/12	2024	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
PICCOLI, MARIA FABIANA	Prof. Responsable	P.Asoc Exc	40 Hs
OCHOA, JULIO CESAR	Responsable de Práctico	JTP Semi	20 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
3 Hs	Hs	2 Hs	3 Hs	8 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
11/03/2024	21/06/2024	15	120

IV - Fundamentación

El presente curso esta destinado a estudiantes de 5to. año de la Lic. en Cs. Computación. La materia asume familiaridad con la programación secuencial y se concentra en técnicas y herramientas para diseñar programas en entornos distribuidos y paralelos.

V - Objetivos / Resultados de Aprendizaje

El objetivo de esta materia es proveer una visión amplia que guíe a estudiantes de la cátedra que deseen trabajar en ambientes distribuidos/paralelos. La materia trata los conceptos más importantes de la temática: características principales de los sistemas distribuidos, discusión para el diseño de algoritmos paralelos aplicando distintas herramientas, análisis de desempeño e implementación de programas paralelos, con numerosos ejemplos que ilustran los principios fundamentales.

Objetivos específicos:

Desarrollar en el alumno la capacidad de conocer y distinguir las características esenciales de sistemas de computación no convencionales; máquinas de alta performance, máquinas paralelas, sistemas distribuidos y otras arquitecturas de uso específico

VI - Contenidos

UNIDAD Nro. 1:

Sistemas Distribuidos y Paralelos - Hardware y Software.

Introducción. Características de Sistemas Distribuidos y de los Sistemas Paralelos. Comparación.

Clasificación de FLYNN. MIMD: Clasificación. Cluster. Grid. Computación Grid.

MIMD de Memoria Compartida y de Memoria Distribuida. Hardware MIMD. Software MIMD. Arquitecturas Multi-core y Many-core.

Arquitectura de Software de los Sistemas Distribuidos.

UNIDAD Nro. 2:

Comunicación en Sistemas Distribuidos. Modelo Cliente- Servidor. Protocolos de direccionamientos. Primitivas de Sincronización Llamadas a procedimiento remotos (RPC).

Objetos distribuidos: Paso de mensajes frente a objetos distribuidos. Sistemas de objetos distribuidos. La arquitectura de Java RMI. API de Java RMI. Comparación entre RMI y la API de sockets.

UNIDAD Nro. 3:

Sistemas Distribuidos. Transacciones Atómicas. Espacio de Trabajo Privado. Writeahead log. Protocolo "Two Phase Commit". Control de Concurrencia. Bloqueo. Control de Bloqueo. Falso Bloqueo. Detección y Prevención de Bloqueo. Algoritmos.

Administración de Procesos y procesadores. Archivos Distribuidos.

UNIDAD Nro. 4:

Programación Paralela, Computación Paralela de Memoria Distribuida y de Memoria Compartida. Paralelismo de Datos y Paralelismo de Tareas. Estructuras típicas de Programación Paralela: Farm, Divide y Vencerás, Map, Iterativo, Compuesto, entre otros.

Medidas de performance de Sistemas de Alto Desempeño. Velocidad. Aceleración. Conjeturas. Aceleración lineal. Gránularidad del Sistema Paralelo.

UNIDAD Nro. 5:

Computación Paralela de Memoria Distribuida: Programación con Pasaje de Mensajes. Creación Dinámica de Procesos. Rutinas básicas de Pasaje de Mensajes.

Concepto de Procesos Demonios. Grupos. Administración de grupos. Tipos de Comunicaciones. Comunicaciones Colectivas. Casos de estudio: MPI.

UNIDAD Nro. 6:

Computación Paralela de Memoria Compartida. Máquinas de Memoria Compartida. MIMD de Memoria Compartida Distribuida (DSM)

Programación de Máquinas de Memoria Compartida. Programación con centinelas (OpenMP) Directivas, Rutinas de Librería y Variables de ambientes. Programación con Threads. Concepto de Thread. Características de los Threads. Threads y Procesos: Similitudes y Diferencias. Beneficios de los Threads: Througput. Desventajas de los Threads. Procesadores Múltiples, Interfaces del Usuario.

UNIDAD Nro. 7:

Computación Paralela Híbrida. Motivación. Características. Ventajas y Desventajas. Características de las aplicaciones. Estrategias de división de problemas. Comunicaciones.

Lenguajes Paralelos. Características Fundamentales. Caso de Estudio: Lenguaje Go.

UNIDAD Nro. 8:

Computación Paralela en Arquitecturas Dedicadas. Motivation. Unidades de Procesamiento Gráfico (GPU). Arquitectura. Computación de propósito General en GPU. CUDA. Modelo de Programación y de Memoria. Ventajas y Desventajas.

UNIDAD Nro. 9:

Que es Big Data? Las 5 Vs de BigData: Volumen, Velocidad, Variedad, Valor, Veracidad. Historia de Big Data (Hadoop, MapReduce, etc). Ciclo de vida de un proyecto Big Data.

UNIDAD Nro. 10:

Que es Green Computing? Características. Necesidades. Relación con Computación de Alto Desempeño.

VII - Plan de Trabajos Prácticos

Práctico 1: Características de Sistemas Distribuidos y paralelos.

Objetivo:

Definir los conceptos de Sistema Distribuido y Sistema Paralelo. Analizar objetivos, similitudes, diferencias, ventajas y desventajas entre computación distribuida y computación paralela. Analizar las diferentes formas en las que el hardware puede ser organizado y qué tipo de organizaciones son adecuadas para los sistemas distribuidos y los sistemas paralelos.

Temas:

Conceptos y generalidades de sistemas distribuidos y paralelos. Clasificación de Flynn. Multiprocesadores y Multicomputadores. Topologías de interconexión.

Metodología:

En este práctico se realizarán una serie de preguntas destinadas a introducir al estudiante dentro de la temática de los sistemas distribuidos y paralelos. Además cada estudiante deberá presentar un informe donde describa claramente una arquitectura paralela real.

Práctico 2: Comunicación en Sistemas Distribuidos.

Objetivo:

Introducir al estudiante a la problemática de la comunicación entre procesos que surge en los sistemas distribuidos. Estudiar y analizar distintos modelos de comunicación. Conducir al estudiante en la lectura de la bibliografía propuesta.

Temas:

El problema de la comunicación entre procesos en los sistemas distribuidos. Modelos de comunicación: Modelo de red, Cliente – Servidor, RPC, Java RMI. Métodos de comunicación: Circuit switching y packet switching.

Metodología:

A través de ejemplos sencillos este práctico introducirá al estudiante en la problemática de la comunicación en los sistemas distribuidos. Este práctico contendrá una guía de preguntas que conducirán al estudiante en el estudio de las particularidades de los distintos modelos de comunicación.

Práctico 3: Transacciones Atómicas.

Objetivo:

Conocer los conceptos y algoritmos fundamentales relacionados con la gestión de transacciones y el control de concurrencia en sistemas distribuidos. Se pretende que el estudiante entienda los conceptos relacionados con la gestión de transacciones centralizadas y distribuidas.

Temas:

Propiedades de las Transacciones. Concurrencia. Transacciones Distribuidas.

Metodología:

En este práctico se plantearán distintas situaciones de transacciones concurrentes y anidadas, debiendo el estudiante analizar las posibles problemas derivados de distintas secuencias de ejecución.

Práctico 4: Análisis de Desempeño de los Sistemas Paralelos

Objetivo:

Analizar un sistema paralelo basándose en distintas medidas de desempeño tal como la aceleración, eficiencia, costo y trabajo. Además que el estudiante sea capaz de criticar las soluciones desarrolladas, justificando sus decisiones de diseño en vista de los resultados de desempeño.

Temas:

Medidas de performance: Aceleración, Eficiencia, Costo y Trabajo. Granularidad del problema.

Metodología:

Este práctico se desarrollará simultáneamente con los prácticos de laboratorio 3, 4, 5 y 6. Por cada uno de los sistemas planteados o desarrollados en los prácticos de laboratorio mencionados, se deberá evaluarlos según los parámetros de rendimiento.

PRÁCTICOS DE LABORATORIO

Práctico 1 Routing en Distintas Topologías.

Objetivo:

Este práctico tiene como objetivo ayudar al estudiante en el repaso de los conceptos adquiridos en teoría acerca de las distintas topologías de interconexión, sus características y particularidades.

Temas:

Topologías de interconexión, routing en distintas topologías, algoritmos de routing.

Metodología:

A través del desarrollo de diferentes programas, se pretende que el estudiante implemente el funcionamiento de distintos algoritmos de routing aptos para las topologías estudiadas.

Práctico 2 Invocación de Métodos Remotos (Java RMI).

Objetivo:

Este práctico presenta una aproximación real al problema de comunicación en los sistemas distribuidos y una posible solución utilizando la tecnología Java RMI. Además se pretende que el estudiante pueda realizar un análisis comparativo entre el modelo utilizado (RMI) y otros modelos tales como CORBA, RPC, etc.

Temas:

Java RMI, Introducción a RPC, Introducción a Corba.

Metodología:

A través del desarrollo guiado de un primer ejercicio, se pretende que el estudiante adquiera las habilidades básicas en el uso de la tecnología Java RMI. Luego el mismo desarrollará una aplicación que solucione un problema descripto por realidad dada, donde sea necesaria la utilización de la tecnología mencionada anteriormente.

Práctico 3: Programación orientada a pasaje de mensajes (MPI).

Objetivo:

Este práctico es una aproximación práctica al diseño e implementación de sistemas paralelos, a fin y efecto que el estudiante sea capaz de resolver los problemas reales aplicando el paradigma de pasaje de mensajes.

Temas:

Programación con Pasaje de Mensajes a través de la biblioteca MPI. Comunicaciones y Sincronizaciones.

Metodología:

A través de distintos problemas simples, algunos resueltos en paralelo y otros a resolver, el estudiante se aproximará en el primer caso a las facilidades provistas por la biblioteca respecto a la programación paralela; y segundo al diseño e implementación de soluciones paralelas a los problemas planteados.

Práctico 4: Programación con Threads

Objetivo:

Este práctico es una aproximación práctica al diseño e implementación de soluciones, a fin y efecto que el estudiante sea capaz de resolver los problemas reales aplicando el paradigma de memoria compartida a través de la programación con threads y las bibliotecas provistas por el sistema operativo.

Temas:

Programación con Threads a través de la biblioteca Pthread. Accesos a Memoria y Sincronizaciones.

Metodología:

A través de distintos problemas simples, algunos resueltos en paralelo y otros a resolver, el estudiante se aproximará en el primer caso a las facilidades provistas por la biblioteca respecto a la programación con threads; y segundo al diseño e implementación de soluciones a los problemas planteados.

Práctico 5 Programación con OpenMP

Objetivo:

Este práctico es una aproximación práctica al diseño e implementación de sistemas paralelos, a fin y efecto que el estudiante sea capaz de resolver los problemas reales aplicando el paradigma de memoria compartida.

Temas:

Programación con Memoria Compartida a través de la biblioteca OpenMP. Directivas. Accesos a Memoria y Sincronizaciones.

Metodología:

A través de distintos problemas simples, algunos resueltos en paralelo y otros a resolver, el estudiante se aproximará en el primer caso a las facilidades provistas por la biblioteca respecto a la programación paralela; y segundo al diseño e implementación de soluciones paralelas a los problemas planteados.

Práctico 6 Programación con OpenMPI

Objetivo:

Este práctico es una aproximación práctica al diseño e implementación de sistemas paralelos, a fin y efecto que el estudiante sea capaz de resolver los problemas reales aplicando el paradigma de programación paralela híbrida.

Temas:

Programación paralela híbrida a través de la biblioteca OpenMPI. Estrategias de división del problema y comunicaciones.

Metodología:

A través de distintos problemas simples, algunos resueltos en paralelo y otros a resolver, el estudiante se aproximará en el primer caso a las facilidades provistas por la biblioteca respecto a la programación paralela; y segundo al diseño e implementación de soluciones paralelas a los problemas planteados.

Práctico 7 Programación CUDA

Objetivo:

Este práctico es una aproximación práctica al diseño e implementación de sistemas paralelos, a fin y efecto que el estudiante sea capaz de resolver los problemas reales aplicando el modelo de programación propuesto para GPU.

Temas:

Programación de GPU a través de la biblioteca CUDA. Funciones kernel. Tipos de Memoria.

Metodología:

A través de distintos problemas simples, algunos resueltos en paralelo y otros a resolver, el estudiante se aproximará en el primer caso a las facilidades provistas por la biblioteca respecto a la programación paralela; y segundo al diseño e implementación de soluciones paralelas a los problemas planteados.

Práctico 8: Trabajo Final

Objetivo:

A través de esta actividad se pretende que el estudiante comprenda la problemática asociada al diseño e implementación de buenos sistemas paralelos, debiendo integrar todo lo visto en la materia, justificando las decisiones de diseño e implementación y los resultados obtenidos.

Temas:

Sistemas Paralelos, integración.

Metodología:

En grupo, de a los más dos personas, los estudiantes desarrollaran un sistema paralela que resuelve un problema real. Cada grupo tendrá un problema real distinto (problemas matemáticos, de ordenamiento, de procesamiento de imágenes, entre otros), para el cual deberán desarrollar una solución paralela, justificando cada una de las decisiones tomadas. Todo lo desarrollado será plasmado en un informe, donde además se analizarán los resultados obtenidos y las posibles mejoras.

VIII - Regimen de Aprobación

a) Realizar los prácticos de aula.

Se entiende por práctico de aula a todo práctico que la cátedra fije para cada unidad, a realizar o entregar en el aula.

b) Aprobar los prácticos de máquina.

Se entiende por práctico de máquina a todo práctico que involucre programación(8 prácticos de laboratorio). Cada práctico se deberá entregar y aprobar en la fecha fijada por la cátedra o en una fecha de recuperación.

También para la aprobación de los mismos se requerirá una asistencia del 80% de los horarios establecidos para prácticos de máquina.

c) Aprobar un examen parcial.

La cátedra establece un global(teórico-práctico) a realizarse al finalizar el dictado. Los estudiantes deberán aprobar tal examen para regularizar, o las dos recuperaciones establecidas en la normativa vigente de la UNSL.

d) Realizar un proyecto de fin de materia

La cátedra fijará el conjunto de aplicaciones paralelas a desarrollar por los estudiantes. El trabajo podrá ser individual o grupal, pudiendo incluir búsqueda bibliográfica y elaboración de monografías. Deberá ser expuesto ante el equipo de cátedra y para toda la clase.

Estarán en condiciones de promocionar la materia aquellos estudiantes que:

- 1- Superen la evaluación continua (llevada a cabo a través del seguimiento de las actividades propuestas en los prácticos de aula y de máquina, como así también en los temas a investigar y exponer)
- 2- Superen las instancias b) y c) con una calificación mayor o igual a 7 (siete)
- 3- Tener una asistencia del 80%.
- 4- Realizar y aprobar el punto d).
- 5- Aprobar un coloquio final sobre todos los conceptos de la materia. El mismo puede ser oral o escrito.

La nota será el promedio de la nota del parcial, del proyecto final, del coloquio y de los prácticos de máquina.

Estarán en condición de regularizar aquellos estudiantes que no cumplan con la evaluación continua, aprueben el parcial con una nota entre 6 y 7, y realicen y aprueben el punto d) y tengan una asistencia de al menos el 70%

EXAMEN FINAL

El examen final será escrito u oral, pudiendo incluir uno o varios temas teóricos y/o prácticos.

EXAMEN LIBRE

Dadas las características prácticas de la materia, los estudiantes libres deberán rendir:

- a) Un examen correspondiente a los prácticos de máquina
- b) Aprobado a) deberá rendir un examen correspondiente a los prácticos de aula.
- c) Aprobado b) rendir el examen oral o escrito, pudiendo incluir uno o varios temas teóricos y/o prácticos.

La no aprobación de alguna de las etapas, implica que el estudiante continúa con la condición libre en la materia.

IX - Bibliografía Básica

- [1] A. Silberschatz, P. Baer Galvin, G. Gagne. Operating system concepts Essentials. 2nd ed. ISBN 9781118804926. Wiley. 2014.
- [2] A. Tanenbaum, M. van Steen, Distributed systems: principles and paradigms. Pearson Prentice Hall, 2007.
- [3] A. Tanenbaum, M. van Steen. Distributed systems : Principles and paradigms. ISBN 0130888931. Prentice-Hall, 2002.
- [4] B. Wilkinson, M. Allen. Parallel Programming: Techniques and Application using Networked Workstations, Prentice-Hall. 1996.
- [5] C. Leopold. Parallel and Distributed Computing: a survey of Models, paradigms and approaches. John Wiley & Son, Inc. 2001.
- [6] D. Kirk, W. Hwu, "Programming Massively Parallel Processors: A Hands-on Approach". ISBN: 978-0-12-381472-2. Elsevier. 2010.
- [7] D. Patterson, John L. Hennessy. Computer organization and design: the hardware / software interface. ISBN 9780128017333. Elsevier. Morgan Kaufmann. 2017.
- [8] G. Coulouris, J. Dollimore, T. Kindberg. Distributed Systems Concepts and Design. Addison Wesley Publishing Company (Second Edition). 2000.
- [9] H. Deitel. Operating Systems. 3th edition. Addison-Wesley. 2004.
- [10] Foster. Designing and Building Parallel Programs. Addison-Wesley. 1994.
- [11] J. Sanders, E. Kandrot, "Cuda by Example: An Introduction to General- Purpose Gpu Programming". ISBN: 0131387685. Addison-Wesley Professional. 2010.
- [12] M. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill Science. 2003.
- [13] M. Quinn. Parallel Computing. Theory and Practice. Second Edition. McGraw-Hill, Inc. 1994.
- [14] M.F. Piccoli, Computación de Alto Desempeño en GPU. ISBN: 978-950-34-0759-2. Edlup. Octubre 2011.

- [15] N. Wilt. The CUDA Handbook: A Comprehensive Guide to GPU Programming. ISBN-10: 0321809467, ISBN-13: 978-0321809469. Addison Wesley, 2013.
- [16] P. Pacheco. Introduction to parallel programming. ISBN 9780123742605. Morgan Kaufmann. Elsevier, 2011.
- [17] Q. Mahmoud. Distributed Programming with Java, Manning, 1999.
- [18] T. Rauber, G. Rünger, Parallel Programming: For Multicore and Cluster Systems, Springer, 2010.
- [19] V. Kumar, A. Grama, G. Karypis, A. Gupta. Introduction to parallel computing: design and analysis of algorithms. Addison Wesley. 2003.
- [20] W. Stallings. Operating systems : Internals and design principles. ISBN 9780132309981. Prentice Hall. 2012.

X - Bibliografía Complementaria

- [1] Apuntes de la Cátedra

XI - Resumen de Objetivos

El presente curso esta destinado a alumnos de 4to. año de la Ing. en Computación. La materia asume familiaridad con la programación secuencial y se concentra en técnicas y herramientas para diseñar programas paralelos.

El objetivo de esta materia es proveer una visión amplia que guíe a estudiantes de la cátedra que deseen construir programas eficientes para computadoras distribuidas y paralelas. La materia trata los conceptos más importantes de la temática y provee una discusión del diseño de algoritmos paralelos, de su análisis de performance y de la construcción de programas, con numerosos ejemplos que ilustran los principios fundamentales.

XII - Resumen del Programa

UNIDAD Nro. 1:

Sistemas Distribuidos y Paralelos - Hardware y Software.

UNIDAD Nro. 2:

Comunicación en Sistemas Distribuidos

UNIDAD Nro. 3:

Sistemas Distribuidos. Transacciones Atómicas.

UNIDAD Nro. 4:

Computación Paralela de Memoria Distribuida y de Memoria Compartida.

UNIDAD Nro. 5:

Computación Paralela de Memoria Distribuida: Programación con Pasaje de Mensajes.

UNIDAD Nro. 6:

Computación Paralela de Memoria Compartida.

UNIDAD Nro. 7:

Computación Paralela Híbrida.

UNIDAD Nro. 8:

Computación Paralela en Arquitecturas Dedicadas (GPU).

UNIDAD Nro 9:

Big Data.

XIII - Imprevistos

La materia se dicta de manera presencial.

Los estudiantes desarrollan los prácticos de máquinas en las salas de máquina y en el cluster de acceso remoto, pudiendo

cumplir con todo lo requerido para la aprobación de la materia.
El seguimiento continuo se realiza a través de la entrega de ejercicios prácticos resueltos.

Contacto: sdyp-unsl@gmail.com

XIV - Otros

--

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	