

Planificación de
Sistemas Operativos II



Código/s: R-412

Identificación y características del Espacio Curricular

Carrera/s:	Licenciatura en Ciencias de la Computación		
Plan de Estudios:	2010, TO2024	Carácter:	Obligatoria
Bloque/Campo:		Área:	Arquitectura, Sistemas Operativos y Redes
Régimen de cursado:	Cuatrimestral		
Cuatrimstre:	7º [LCC], 7º [LCC]		
Carga horaria:	105 hs. / 7 hs. semanales	Formato curricular:	Asignatura
Escuela:	Ciencias Exactas y Naturales	Departamento:	Ciencias de la Computación
Docente responsable:	RUIZ, Esteban		

Programa Sintético

Comunicación. Sincronización. Sockets y protocolos de comunicación. Planificación de procesos. Scheduling. Sistemas de archivos. Políticas de alojamiento y administración de memoria. Sistemas distribuidos. Seguridad y autenticación.

Espacios Curriculares Relacionados

Previos Aprobados: R-222 - Arquitectura del Computador, R-314 - Examen de Suficiencia de Inglés, R-313 - Sistemas Operativos I

Simultaneos Recomendados:

Posteriores:

Vigencia desde 2024

Firma Profesor

Fecha

Firma Aprob. Escuela

Fecha

Con el aval del Consejo Asesor:

Fundamentación

Sistemas Operativos II es una asignatura del cuarto año de la carrera, dictada en el primer cuatrimestre. Presenta los conceptos fundamentales de los sistemas operativos y nociones de diseño. Además se presentan distintos algoritmos, políticas y heurísticas que pueden aplicarse a diversos problemas computacionales. Permite comprender la relación entre el hardware y los programas de usuario, solucionar problemas u optimizar el funcionamiento de un sistema de cómputo. Dentro del campo de la investigación, diversas partes de un sistema operativo (e incluso su diseño general) admiten y muchas veces requieren modificaciones y optimizaciones para adaptarse a cambios en el software, en el hardware o en el entorno dando lugar a muchas alternativas de investigación y desarrollo. Los diversos campos de aplicación de un sistema operativo y el advenimiento de la computación en paralelo y heterogénea crean además nuevos desafíos en el campo de la investigación. Esta materia sirve de base para comprender otros conceptos como aspectos de seguridad informática, bases de datos y compiladores y brinda conceptos fundamentales para realizar auditorías, análisis y evaluaciones de proyectos de software. También se forma al alumno con fundamentos de comunicación efectiva y trabajo en equipo que contribuyen a las finalidades de dictar clases, comunicarse convenientemente en entornos laborales e interactuar con otros profesionales. La capacitación en el desarrollo de proyectos de software brinda herramientas para gestionar o desempeñarse en proyectos de software de diferente envergadura.

Resultados del aprendizaje

Al finalizar el cursado los/las estudiantes serán capaces de:

RA1 Escoger y parametrizar distintas políticas y opciones en las distintas partes de un sistema operativo: procesos, memoria virtual y sistemas de archivos
RA2 Analizar el diseño de las principales partes de un sistema operativo: planificación, memoria virtual, sistemas de archivos, seguridad
RA3 Identificar y resolver problemas en sistemas operativos existentes y proponer optimizaciones.
RA4 Reconocer los nuevos avances y tendencias en el área de sistemas operativos
RA5 Realizar en equipo en forma organizada y autogestionada el desarrollo de un proyecto de informática complejo y de larga duración, cumpliendo roles determinados e interactuando con el resto del equipo de forma eficiente.
RA6 Comunicar en forma efectiva el progreso y los resultados de los trabajos realizados.

Competencias / Ejes transversales y Resultados del Aprendizaje

Competencia/Eje transversal al que tributa	Nivel	Resultados del Aprendizaje
CGT1-Identificación, formulación y resolución de problemas de informática	Alto	RA1, RA2, RA3, RA4, RA5
CGT2-Concepción, diseño y desarrollo de proyectos de informática	Alto	RA1, RA4, RA5, RA6
CGT3-Gestión, planificación, ejecución y control de proyectos de informática	Bajo	RA1, RA4
CGT4-Utilización de técnicas y herramientas de aplicación en la informática	Bajo	RA1, RA4
CGT5-Generación de desarrollos tecnológicos y/o innovaciones tecnológicas	Bajo	RA1, RA4
CGS1-Fundamentos para el desempeño en equipos de trabajo	Alto	RA5
CGS2-Fundamentos para la comunicación efectiva	Medio	RA6

Programa Analítico

Unidad 1

Introducción a los Sistemas Operativos

- 1.1 Introducción. Historia y propósitos de los sistemas operativos.
- 1.2 Batch, Time-Sharing
- 1.3 Arquitecturas generales.
- 1.4 Hitos fundamentales
- 1.5 Desempeño en equipos de trabajo: autogestión y organización.

Unidad 2

Procesos

- 2.1 Procesos y threads
- 2.2 Modos usuario y supervisor. Traps. Etapas y regiones de un proceso.
- 2.3 Primitivas de creación y terminación.
- 2.4 Comunicación efectiva: creación de informes, organización de la información a presentar.
- 2.5 Desarrollo de proyectos de informática: gestión de versiones, análisis, programación y verificación

Unidad 3

Comunicación entre procesos

- 3.1 Comunicaciones interprocesos. Sincronización.
- 3.2 Abstracciones: streams, packets. Protocolos: TCP/IP y sockets

Unidad 4

Multitarea y tiempo real

- 4.1 Multitarea
- 4.2 Políticas de scheduling. FCFS, SJF, prioridades, etc
- 4.3 Tiempo Real y problemas asociados
- 4.4 Comunicación efectiva de conceptos y heurísticas. Utilización de roles en equipos de trabajo.
- 4.5 Desarrollo de proyectos de informática: metodologías, ciclo de vida

Unidad 5

Sistemas de archivos

- 5.1 Sistemas de archivos. Data y metadata.
- 5.2 Distintas organizaciones de metadata
- 5.3 Inodos. Fast File System y Ext2. Journaling
- 5.4 Fundamentos para la comunicación efectiva del trabajo realizado y presentación de trabajos finales

Unidad 6

Nociones de seguridad

- 6.1 Seguridad

Unidad 7

Entrada y salida

- 7.1 Sistemas de entrada/salida
- 7.2 Coordinación y caches. Abstracciones. Servicios adicionales

Unidad 8

Memoria virtual

- 8.1 Memoria física y virtual.
- 8.2 Espacios de direcciones. Segmentación y paginación.
- 8.3 Políticas de desalojo, FIFO y anomalía de Belady. Propiedad de pila
- 8.4 LRU. Aproximaciones al LRU. Dirty bit y Second Chance

Unidad 9

Privacidad y encriptación

- 9.1 Privacidad y seguridad
- 9.2 Sistemas de encriptación. Sistemas de clave secreta y de clave pública

Unidad 10

Tendencias actuales y tópicos avanzados

10.1 Tendencias actuales y posibles tendencias futuras

10.2 Microkernels, exokernels, etc.

Modalidades de enseñanza

Clases teóricas presenciales en aula o laboratorio acompañadas con presentación audiovisual y utilización de pizarra y equipos de cómputo.

Clases prácticas presenciales en laboratorio presentando ejercicios de aplicación de los conceptos estudiados en modalidad de taller.

Trabajos prácticos grupales basados en la resolución de problemas de computación y en la aplicación de los contenidos dictados. Estos trabajos se desarrollarán en el laboratorio de informática con la guía de los docentes, así como también fuera del horario de clases fomentando el autoaprendizaje y el trabajo en equipo. Durante la mayor parte del cuatrimestre se utiliza el sistema operativo educacional Nachos que provee el esqueleto de un sistema operativo que los alumnos deben completar tomando decisiones de diseño, implementación y rendimiento y haciendo un análisis de las distintas alternativas.

Lectura de temas por parte de los alumnos y posterior discusión en clases (para detallar los puntos más importantes), utilizando bibliografía previamente asignada por los docentes.

Explicación o detalles de los conceptos teóricos aplicados en un sistema operativo real (principalmente GNU/Linux)

Recursos

Se utiliza un laboratorio con proyector multimedia y computadoras con GNU/Linux para exponer y realizar las prácticas. Se utiliza el campus virtual de la facultad como recurso primario para toda la información disponible para el alumno, registrar el resultado de las evaluaciones, enviar comunicaciones, determinadas entregas, etc. (<https://campusv.fceia.unr.edu.ar/>). Además el repositorio de control de versiones svn (<https://svn.dcc.fceia.unr.edu.ar/svn/lcc/R-412/>) se utiliza como repositorio de material bibliográfico, software y para que los alumnos realicen el versionado del código fuentes entregas de software. También es utilizado por los docentes para almacenar información de la cátedra (código fuente de apuntes, materiales, notas, actas, etc...). La página web de la lcc tiene además información resumida sobre la materia: <https://dcc.fceia.unr.edu.ar/es/lcc/r412> . Además el servidor de desarrollo para alumnos de la carrera (labdcc.fceia.unr.edu.ar) permite mostrar distintas configuraciones del sistema.

El software NACHOS (<https://homes.cs.washington.edu/~tom/nachos>) (con modificaciones de la Universidad de Gran Canaria y modificaciones propias - <https://svn.dcc.fceia.unr.edu.ar/svn/lcc/R-412/Public/nachos/> -) se utiliza como punto de partida para la mayor parte de las prácticas. Otro software utilizado incluye: svn, gcc, make, gdb, valgrind, utilitarios de línea de comandos del shell de GNU/Linux (grep, find, cut, cat ...) y específicos para demostrar alguna característica (fsdb, fsck, mkfs, filesystem /proc, dmesg, top, time).

Actividades de Formación Práctica

Trabajos prácticos basados en el software educacional NACHOS. NACHOS es un Sistema Operativo educativo para los estudiantes de cursos de Sistemas Operativos, sin graduación o aspirantes a graduados. Fue desarrollado en la Universidad de California en Berkeley por Wayne A. Christopher, Steven J. Procter, y Thomas E. Anderson entre 1991 y la primavera de 1992, y es usado por numerosas escuelas. Escrito originalmente en C++ para MIPS, NACHOS se ejecuta como un proceso de usuario en el sistema operativo anfitrión. Un simulador de MIPS (incluido con el software) ejecuta el código de sus programas de usuario. La cátedra modificó sustancialmente las prácticas previstas en el software original.

Nº	Título	Descripción
----	--------	-------------

1	Introducción a Nachos	Ejercicios de Nachos, presenta la arquitectura general del sistema y permite familiarizarse con la estructura del código y las principales partes de un sistema operativo.
2	Sincronización de hilos	Ejercicios de Nachos, incluye la implementación de primitivas de concurrencia, sincronización y modificaciones al planificador
3	Programas de usuario y multiprogramación	Ejercicios de Nachos, incluye la implementación de llamadas al sistema y la gestión de la multiprogramación.
4	Sistema de archivos	Ejercicios de Nachos, mejoras al sistema de archivos existente (basado en el UNIX FS): soporte multiusuario, tamaño variable, archivos grandes y subdirectorios.
5	Memoria virtual	Ejercicios de Nachos, avanza sobre el uso de la TLB por parte del sistema operativo, carga por demanda y políticas de paginación.
6	Sistemas distribuidos, virtualización y micronúcleos	Cuestionario teórico sobre los últimos temas

Evaluación

A lo largo del cuatrimestre se realizan diversas prácticas y se toman dos exámenes parciales. Para regularizar la asignatura se deben entregar y aprobar el 80% de las prácticas en las fechas convenidas y aprobar los dos parciales con nota mínima seis en cada uno de ellos. Se cuenta con una instancia recuperatoria al final del cuatrimestre. Se evalúan además en forma cualitativa y en forma continua el aprendizaje de la distintas competencias utilizando un esquema basado en objetivos logrados o no logrados para cada competencia. El examen final incluye la exposición oral del trabajo realizado por el grupo (revisión de carpeta). La nota final se calcula en función de los exámenes parciales, entregas realizadas durante el año y la nota del examen final.

Resultado de Aprendizaje	Actividades/Modalidad de Enseñanza	Modalidad de Evaluación
RA1	Clases teóricas presenciales en aula con presentación audiovisual. Clases prácticas presenciales en laboratorio resolviendo ejercicios de aplicación de los conceptos estudiados bajo la supervisión y ayuda de los docentes. Trabajos prácticos grupales basados en la resolución de problemas de computación.	Exámenes parciales Trabajos prácticos Examen final
RA2	Clases teóricas presenciales en aula con presentación audiovisual. Clases prácticas presenciales en laboratorio resolviendo ejercicios de aplicación de los conceptos estudiados bajo la supervisión y ayuda de los docentes. Trabajos prácticos grupales basados en la resolución de problemas de computación.	Exámenes parciales Trabajos prácticos Examen final

RA3	Clases teóricas presenciales en aula con presentación audiovisual. Clases prácticas presenciales en laboratorio resolviendo ejercicios de aplicación de los conceptos estudiados bajo la supervisión y ayuda de los docentes. Trabajos prácticos grupales basados en la resolución de problemas de computación.	Exámenes parciales Trabajos prácticos Evaluación continua Examen final
RA4	Clases teóricas presenciales en aula con presentación audiovisual. Clases prácticas presenciales en laboratorio resolviendo ejercicios de aplicación de los conceptos estudiados bajo la supervisión y ayuda de los docentes. Trabajos prácticos grupales basados en la resolución de problemas de computación.	Trabajos prácticos Examen final
RA5	Clases teóricas presenciales en aula con presentación audiovisual. Trabajos prácticos grupales basados en la resolución de problemas de computación.	Trabajos prácticos Evaluación continua Examen final
RA6	Clases teóricas presenciales en aula con presentación audiovisual.	Trabajos prácticos Evaluación continua Examen final

Bibliografía básica

Autores (Apellido, Inicial nombre)	Año de edición	Título de la obra	Editorial o Revista	Ejemplares disponibles o sitio web
Wolf, Gunnar; Ruiz, Esteban; Bergero, Federico y Meza, Erwin	2015	Fundamentos de sistemas operativos	Universidad Nacional Autónoma de México	https://ru.iiec.unam.mx/2718/
William Stallings	2011	Operating systems: internals and design principles. 7th ed	Pearson	1
Andrew S. Tanenbaum, Albert S. Woodhull	2006	Operating Systems: Design and Implementation. 3rd ed	Pearson	1
Tanenbaum	2009	Sistemas Operativos Modernos	Pearson	2
Tanenbaum	1999	Sistemas Operativos. Diseño e implementación	Pearson	3
Silberschatz, Galvin, Gagne	2010	Fundamentos de Sistemas Operativos	Mc. Graw Hill	1

Bibliografía complementaria

Autores (Apellido, Inicial nombre)	Año de edición	Título de la obra	Editorial o Revista	Ejemplares disponibles o sitio web
Tanenbaum		Sistemas Operativos Distribuidos	Prentice Hall & IBD	1

Distribución de la carga horaria**Presenciales**

Teóricas				40 Hs.
Prácticas			Formación Experimental	
			Resolución de Problemas vinculados a la Profesión	20 Hs.
			Resolución de Problemas y Ejercicios	20 Hs.
			Actividades de Proyecto y Diseño	20 Hs.
		Formación en la Práctica Profesional		
Evaluaciones				5 Hs.
			Total	105 Hs.

Dedicadas por el alumno fuera de clase

			Preparación Teórica	35 Hs.
			Preparación Práctica	40 Hs.
			Elaboración y redacción de informes, trabajos, presentaciones, etc.	30 Hs.
			Total	105 Hs.

Cronograma de actividades

Semana	Unidad	Tema	Actividad
1	1	1.1 Introducción. Historia y propósitos de los sistemas operativos 1.2 Batch, Time-Sharing, distribuidos 1.1 Introducción. Historia y propósitos de los sistemas operativos 1.2 Batch, Time-Sharing, distribuidos 1.3 Arquitecturas generales	Clases teóricas y prácticas.
2	1 y 2	1.4 Hitos fundamentales 1.5 Desempeño en equipos de trabajo: autogestión y organización. 2.1 Procesos y threads 2.2 Modos usuario y supervisor. Traps. Etapas y regiones de un proceso. 2.3 Primitivas de creación y terminación. Operaciones. 2.4 Comunicación efectiva: creación de informes, organización de la información a presentar. 2.5 Desarrollo de proyectos de informática: gestión de versiones, análisis, programación y verificación	Clases teóricas y prácticas. Desarrollo de ejemplos.

3	3 y 4	3.1 Comunicaciones interprocesos. Sincronización. 3.2 Abstracciones: streams, packets. Protocolos: TCP/IP y sockets 4.1 Multitarea	Clases teóricas y prácticas. Desarrollo de ejemplos. Entrega 1
4	4	4.2 Políticas de scheduling. FCFS, SJF, prioridades, etc	Clases teóricas y prácticas. Desarrollo de ejemplos.
5	4	4.2 Políticas de scheduling. FCFS, SJF, prioridades, etc	Clases teóricas y prácticas. Desarrollo de ejemplos.
6	4	4.3 Tiempo Real y problemas asociados 4.4 Comunicación efectiva de conceptos y heurísticas. Utilización de roles en equipos de trabajo. 4.5 Desarrollo de proyectos de informática: metodologías, ciclo de vida	Clases teóricas y prácticas. Desarrollo de ejemplos.
7	5	5.1 Sistemas de archivos. Data y metadata. 5.2 Distintas organizaciones de metadata	Clases teóricas y prácticas. Desarrollo de ejemplos.
8	5	5.3 Inodos. Fast File System y Ext2. Journaling	Clases teóricas y prácticas. Desarrollo de ejemplos. Entrega 2 (userprog)
9	5	5.3 Inodos. Fast File System y Ext2. Journaling 5.4 Sistemas de archivos remotos 5.4 Fundamentos para la comunicación efectiva del trabajo realizado y presentación de trabajos finales	Clases teóricas y prácticas. Desarrollo de ejemplos. Parcial 1
10	6 y 7	6.1 Seguridad 7.1 Sistemas de entrada/salida 7.2 Coordinación y caches. Abstracciones. Servicios adicionales	Clases teóricas y prácticas. Desarrollo de ejemplos.
11	8	8.1 Memoria física y virtual. 8.2 Espacios de direcciones. Segmentación y paginación.	Clases teóricas y prácticas. Desarrollo de ejemplos. Entrega 3 (filesystem)
12	8	8.3 Políticas de desalojo, FIFO y anomalía de Belady. Propiedad de pila 8.4 LRU. Aproximaciones al LRU. Dirty bit y Second Chance	Clases teóricas y prácticas. Desarrollo de ejemplos.
13	9	9.1 Privacidad y seguridad 9.2 Sistemas de encriptación. Sistemas de clave secreta y de clave pública	Clases teóricas y prácticas. Desarrollo de ejemplos. Parcial 2
14	10	10.1 Tendencias actuales y posibles tendencias futuras	Clases teóricas y prácticas. Entrega 4 (vmem)
15	10	10.2 Microkernels, exokernels, etc.	Clases teóricas. Entrega 5 (cuestionario), Recuperatorio