

EX-2026-00509716- -UNC-ME#FAMAF

PROGRAMA DE ASIGNATURA	
ASIGNATURA: Sistemas Operativos	AÑO: 2026
CARÁCTER: Obligatoria	UBICACIÓN EN LA CARRERA: 2° año 2° cuatrimestre
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: Cuatrimestral	CARGA HORARIA: 120 horas

FUNDAMENTOS Y OBJETIVOS

Fundamentación: El sistema operativo es un programa fundamental dentro de toda la pila de software y hardware que compone una computadora moderna. Esto es así no solamente porque aísla a los programas de usuario de los detalles del hardware subyacente, sino que además provee fuertes abstracciones que han perdurado a lo largo de las décadas: procesos, memoria (virtual) y sistema de archivos.

La necesidad de aprovechar mejor el hardware hizo que apareciera el concepto de multiprogramación, donde el no-determinismo de los programas secuenciales se presenta de manera concreta, además de presentar el Área de la Teoría y la Práctica de la Concurrencia.

Objetivos:**Teórico**

- Comprender las abstracciones principales de un Sistema Operativo: procesos, memoria, sistema de archivos.
- Resolver problemas simples que se plantean en la práctica para estas abstracciones.
- Comprender, reparar y programar algoritmos concurrentes de baja complejidad.
- Entender la problemática de la seguridad en general, y para los Sistemas Operativos en particular.
- Resolver problemas sencillos que involucren algunos de los aspectos sobresalientes de la seguridad y la entrada/salida en sistemas operativos.
- Entender las relaciones de compromiso de los algoritmos y estructuras de datos internas del Sistema Operativo. Comprender cómo algunos cambios tecnológicos afectan fuertemente estas relaciones de compromiso.
- Comprender la relación entre algunas partes del diseño de la arquitectura del microprocesador con el Sistema Operativo.
- Poder asimilar los conceptos utilizando ejemplos concretos de Sistemas Operativos.

Laboratorio

- Avanzar en la práctica de la programación en general.
- Trabajar en grupo tanto en objetivos individuales como en objetivos grupales.
- Ser capaz de leer, modificar y comprobar código dentro de Sistemas Operativos completos y funcionales.
- Utilizar herramientas de apoyo para el desarrollo del software: editores, detectores de errores en código estático, debuggers, chequeadores de memoria, etc.
- Utilizar herramientas de desarrollo colaborativo de proyectos y las prácticas asociadas a desarrollos remotos.
- Generar independencia para la búsqueda de soluciones técnicas en el proceso de desarrollo y/o modificación de código.
- Realizar entregas de proyectos dentro de límites de tiempo prefijados.
- Programar abstracciones de dispositivos de bajo nivel a partir de la especificación dada por su hoja de datos.
- Realizar modificaciones a partes fundamentales del Sistema Operativo: procesos, memoria

EX-2026-00509716- -UNC-ME#FAMAF

virtual y sistema de archivos.

- Comprender en general la problemática del desarrollo del software dentro del núcleo del Sistema Operativo.

El teórico de la materia será clase expositiva tradicional con mezcla de live-coding dependiendo de tener un beamer en la sala de dictado.

Los laboratorios serán presenciales.

CONTENIDO

1 - Virtualización

Virtualización de CPU: procesos, API de procesos, ejecución directa limitada, planificación, planificación multinivel. Virtualización de RAM: espacio de direcciones, API de memoria, traducción de direcciones, segmentación, administración de memoria libre, introducción a la paginación, TLB, tablas de páginas avanzadas, archivo de intercambio.

2 - Concurrencia

Concurrencia e hilos, API de hilos, locks, variables de condición, semáforos, bugs de concurrencia.

3 - Persistencia

Dispositivos de Entrada/Salida, discos duros rotacionales, RAID, archivos y directorios, implementación de sistemas de archivos, sistemas de archivos rápidos, fsck y bitácora, sistemas de archivos con registro.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Remzi Arpaci-Dusseau, Andrea C. Arpaci-Dusseau, Operating Systems: Three Easy Pieces, University of Wisconsin-Madison, November, 2023 (Version 1.10), <https://pages.cs.wisc.edu/~remzi/OSTEP/#book-chapters>
- Andrew S. Tanenbaum, Herbert Bos, Sistemas Operativos Modernos, Quinta Edición. Prentice Hall, 2022.
- Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Operating System Concepts 10th Edition, Wiley, 2021.

BIBLIOGRAFÍA COMPLEMENTARIA

- Michael Kerrisk, The Linux Programming Interface, No Starch Press, 2010.
- Russ Cox, Frans Kaashoek, Robert Morris, xv6 a simple, Unix-like teaching operating system, MIT, 2012, <https://pdos.csail.mit.edu/6.828/2023/xv6/book-riscv-rev3.pdf>
- Raphael Finkel. An operating systems Vade Mecum, Segunda Edición. Prentice Hall, 1988.
- Gunnar Wolf, Esteban Ruiz, Federico Bergero, Erwin Meza. Fundamentos de Sistemas Operativos, 2015.
- Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman. Linux Device Drivers, Third Edition. O'Reilly, 2005.

METODOLOGÍA DE ENSEÑANZA

La secuenciación de contenidos se estructura alrededor del libro OSTEP de la materia: virtualización, concurrencia y persistencia. Los laboratorios siguen esta secuencia de manera exacta, aunque no siempre sincronizada con el teórico.

EX-2026-00509716- -UNC-ME#FAMAF

Los tiempos de las 3 partes se dividen en dos, hasta el primer parcial que ocurre en la semana del estudiante que incluye el 21 de septiembre entra virtualización y para el parcial final en la 3ra semana de noviembre concurrencia y virtualización. Este aparente desbalance sirve para empezar lento y luego ir acelerando a medida que las/os estudiantes se afianzan en la lógica de la materia, que es la primera que tienen donde se mezclan conocimientos fundamentales (concurrencia), con tecnologías y formas de su aplicación (virtualización y persistencia).

Las clases teóricas son expositivas con prácticas en papel para realizar y que se van resolviendo en un teórico-práctico que ocurre en 4 de las 8hs semanales disponibles. El laboratorio se estructura en base a un proyecto introductorio a la materia y de preparación (Lab0) y tres laboratorios uno por tema. En el laboratorio se presentan los proyectos y luego el cuerpo docente orienta el trabajo de los grupos, integrados por 4 estudiantes. Como parte de la gestión de las clases, se realizan presentaciones sobre manejo de herramientas que son necesarias para el laboratorio (frameworks de testing, sistemas de control de versiones, etc.)

Las actividades en los teóricos se basan en la resolución de problemas en papel y lápiz o sea las guías de práctico. Además, se van compartiendo de manera regular noticias absolutamente actuales sobre las diferentes temáticas que se van tocando, a fin que las/os alumnos/as entiendan que el conocimiento de Sistemas Operativos es actual y tiene implicancias sociales. Se promueve el uso del canal de Zulip #sistop-2026 a fin de discutir problemáticas, dudas, ejercicios y todo lo que compete a la materia.

En los laboratorios las actividades son la compleción de proyectos donde se presenta una pieza de software y cada grupo debe lograr extenderla a fin de cumplir con los objetivos del laboratorio. Cada laboratorio tiene casos de testing bien definidos. Los proyectos de laboratorio incluyen la realización de un video de 7 minutos donde el grupo debe exponer dificultades y aprendizajes del proceso. Es parte de los materiales entregados los commits en el repositorio de cada grupo con pautas claras acerca de cómo se deben realizar estos commits.

Los recursos didácticos del teórico se basan principalmente en material generado a lo largo de los años por estudiantes y docentes. En este sentido, la plataforma Zulip funciona como repositorio de todos estos materiales.

Para el laboratorio además de Zulip se cuenta con Bitbucket para la administración de versiones y material seleccionado a lo largo de más de 15 años en formatos de guías, videos y guías de estudio que se acceden a través de la plataforma Moodle.

La dinámica de trabajo del teórico es clásica, de tipo clase magistral, con la búsqueda de interacción a partir de los conocimientos o preconceptos de las y los estudiantes. En los laboratorios se promueve el trabajo colaborativo y autónomo al interior de los grupos. El cuerpo docente interviene para atender las dudas e inquietudes que se presentan a medida que los grupos avanzan con sus producciones.

EVALUACIÓN

FORMAS DE EVALUACIÓN

- Dos parciales teórico-práctico presenciales.
- Cuatro (4) proyectos grupales. Todos los proyectos permiten una re-entrega.
- Régimen de promoción.
- Examen final teórico-práctico presencial, más coloquio sobre los Laboratorios si el alumno está libre.

REGULARIDAD

Para regularizar la materia se necesita aprobar todos los Laboratorios con 6 o más. Los

EX-2026-00509716- -UNC-ME#FAMAF

laboratorios se corrigen solo por resultado de ejecuciones ya que la escala de la cursada no permite revisión de código.

PROMOCIÓN

Para promocionar la materia se necesita: Aprobar los dos parciales de teórico con un promedio de 7 (cada parcial con más de 6). Aprobar todos los laboratorios y sus parciales con 6 o más.

En caso de promocionar la calificación final será

$$\text{notaPromoción} = \min(10, 0.35 * p1 + 0.35 * p2 + 0.4 * \text{lab})$$

donde p1 y p2 son las calificaciones del primer y segundo parcial respectivamente, y lab es la calificación del desempeño en los laboratorios. Todas las calificaciones van entre 0 y 10.

Cualquier estudiante puede rendir libre la materia. En este caso deberá presentar 14 días antes de la fecha de examen todos los laboratorios completos. Primero se tomará un examen en máquina con ejercicios relacionados a los laboratorios, luego el/la estudiante deberá defender sus laboratorios en un coloquio a posteriori de que haya aprobado el examen en máquina. Luego de pasar estas dos instancias podrá rendir el examen teórico-práctico de la materia.