

## PROGRAMA de Sistemas Operativos

**Carrera/s:** Tecnicatura Universitaria en Programación Informática - Licenciatura en Informática

**Asignatura:** Sistemas Operativos

**Núcleo al que pertenece:** Básico

**Profesor/es:** Juan Marcelo Tondato, Fernando García, Esteban Dimitroff y Federico Repond

**Asignaturas Correlativas:** Introducción a la Programación – Organización de Computadoras

### Objetivos:

Que el estudiantado:

- Conozca las funciones de un sistema operativo, las características de los recursos que administra y las diferentes políticas de administración.
- Conozca los conceptos de proceso y thread, la necesidad de planificar y controlar su ejecución, y tenga nociones de políticas de planificación.
- Se familiarice con los entornos operativos del estilo Unix, el uso de interfaz de línea de comandos, y adquiera conocimientos sobre shell scripting.

### Contenidos mínimos:

- Introducción a los sistemas operativos: función de abstracción del hardware; organización, estructura y servicios de los SO. Tipos de sistemas (Sistemas batch / Multiprogramación / Sistemas de tiempo real / Sistemas distribuidos / Sistemas paralelos / Sistemas embebidos).
- Procesos y threads: Conceptos de proceso, thread y planificación. Comunicación y cooperación entre procesos. Deadlocks.
- Planificación: Algoritmos, criterios. Multiprocesamiento.
- Manejo de memoria: Espacio lógico vs físico, swapping, alocaión contigua, paginación, segmentación.
- Memoria virtual: Paginación bajo demanda, algoritmos de reemplazo de página, thrashing.
- Sistemas de archivos: Manejo de archivos, manejo de directorios.
- Protección: objetivos, dominio de protección, matriz de acceso y sus implementaciones.
- Prácticas, trabajos incluyendo uso de shell scripting e instalaciones en distintos sistemas operativos, en particular del estilo Unix: GNU/Linux, etc.

**Carga horaria semanal:** 6 hs

**Programa analítico:**

**Unidad 1: Introducción**

Gestión (eficiente) de recursos y abstracción del hardware. Revisión del Hardware manejado por un SO: procesador, memoria, dispositivos de E/S, buses. System Calls. Estructura de un SO.

**Unidad 2: Procesos y threads**

Comunicación entre procesos. Planificadores de Procesador. Políticas de Asignación de Procesador. Algoritmos. Concepto de 'Working Set'. Sincronización. Semáforos. Multiprocesamiento. Procesos Concurrentes y Distribuidos.

**Unidad 3: Deadlocks**

Condiciones Necesarias. Grafos de Asignación de Recursos. Formas de Evitar el Abrazo Mortal. Algoritmo del Banquero. Algoritmos de Detección. Recuperación: selección de víctimas. Inanición.

**Unidad 4: Administración de memoria**

Particionamiento fijo y variable. Particionada variable reubicable (con compactación). Paginación. Swapping. Overlays. Segmentación. Segmentación paginada.

**Unidad 5: Memoria Virtual**

Concepto de memoria virtual. Paginación bajo demanda. Análisis de Algoritmos de Reemplazo de Página. Asignación de Páginas. Trashing.

**Unidad 6: Sistemas de archivos**

Manejo de archivos. Manejo de directorios. Implementación de sistemas de archivos. Ejemplos de sistemas existentes.

**Unidad 7: Administración de E/S**

Buses. DMA. Unidad de control de periféricos. Dispositivos dedicados y compartidos. Dispositivos virtuales (SPOOLING). Planificación de Entrada/Salida. Controlador de tráfico. Manipulador de periféricos.

**Unidad 8: Protección y Seguridad**

Dominio de Protección. Matriz de Acceso. Sistema de Capacidades. Validación. Contraseñas. Amenazas. Encriptación. Clasificación.

**Bibliografía obligatoria:**

- Silberschatz. Fundamentos de Sistemas Operativos. McGraw-Hill. 2006
- Tanenbaum. Sistemas Operativos Modernos. Prentice Hall. 1996
- Tanenbaum. Sistemas Operativos - Diseño e Implementación. Prentice Hall. 1988

**Bibliografía de consulta:**

- Cameron Newham, Bill Rosenblatt. Learning the Bash Shell. O'Reilly & Associates, Inc., Sebastopol, CA, USA. 1998

- Brett Slatkin. *Effective Python: 59 Specific Ways to Write Better Python* (1st ed.). Addison-Wesley Professional. 2015

### **Organización de las clases:**

Se desarrollarán clases teóricas y prácticas. El objetivo de las clases prácticas es implementar un trabajo práctico que incluya los temas más relevantes de la teoría.

Las actividades extra-áulicas incluyen la lectura para repaso de los temas introducidos en la teoría y desarrollo del simulador para llegar a las clases prácticas con consultas relevantes que permitan discutir el diseño y próximos pasos.

### **Trabajos prácticos a desarrollar:**

- Trabajo Práctico 1: Introducción, componentes, Kernel, System Calls: Se solicita a quienes cursen la materia, que estudie y describa cómo funciona un emulador de Sistema Operativos entregado como base para la realización de los Tps. Se entrega un emulador de un SISTEMA OPERATIVO y un HARDWARE subyacente (ej: CPU, MEMORIA, Vector de interrupciones y otros componentes básicos) programado en python para usar como base de los TPs.

Bibliografía de soporte: Capítulo 1: Introducción – Tanenbaum

- Trabajo Práctico 2: Sistemas BATCH: Quien curse debe evolucionar su Sistema Operativo para que corra varios procesos por LOTES o BATCH.

Bibliografía de soporte: Capítulo 2: Procesos y Threads - Tanenbaum

- Trabajo Práctico 3: PCB y procesos. El objetivo de este TP es que quien curse la materia, evolucione su Sistema Operativo para poder correr varios procesos al mismo tiempo, para esto es necesario el desarrollo de las estructuras de datos y manejo de IRQ correspondientes a la problemática a resolver.

Bibliografía de soporte: Capítulo 2: Procesos y Threads - Tanenbaum

- Trabajo Práctico 4: Planificadores de Procesador: Políticas de Asignación de Procesador. Algoritmos. Una vez resuelto el problema de multiprogramación, se exploran y analizan los distintos algoritmos, propuestos en la teoría, para maximizar el uso del CPU.

Bibliografía de soporte: Capítulo 2: Procesos y Threads - Tanenbaum

- Trabajo Práctico 5: Particionamiento fijo y variable. Particionada variable reubicable (con compactación). Paginación. El problema ahora se centra en proporcionar un uso eficiente de la Memoria Física, en este caso se solicita al estudiantado que implemente los mecanismos de administración de memoria que se ven en la teoría

Bibliografía de soporte: Capítulo 3: Administración de Memoria – Tanenbaum

- Trabajo Práctico 6: Concepto de memoria virtual. Swapping. Paginación bajo demanda. Análisis de Algoritmos de Reemplazo de Página. Asignación de Páginas. Trashing. Quienes cursen deben agregar a su Sistema Operativo la capacidad de

administrar una partición de Swap y realizar el reemplazo de páginas de memoria a demanda.

Bibliografía de soporte: Capítulo 3: Administración de Memoria – Tanenbaum

- Trabajo Práctico 7: Manejo de archivos. Manejo de directorios. Implementación de sistemas de archivos. Se incorpora un File System al Sistema Operativo.

Bibliografía: Capítulo 4: Sistemas de Archivo – Tanenbaum

**Modalidad de evaluación:**

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los siguientes artículos del Régimen de estudios de la UNQ (Res. CS 201/18)

En la modalidad de libre, se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral e instancias de evaluación similares a las realizadas en la modalidad presencial.



### CRONOGRAMA TENTATIVO

Semana	Tema/unidad	Actividad*			Evaluación	
		Teórico	Práctico			
			Res Prob.	Lab.		Otros Especificar
1	Introducción	x		x		
2	Procesos	x		x		
3	Procesos	x		x		
4	Threads	x		x		
5	Planificación de Procesos	x		x		
6	Planificación de Procesos	x		x		
7	Deadlocks	x		x		
8	Administración de Memoria	x		x		
9	Memoria Virtual	x		x		
10	Memoria Virtual	x		x		
11	Interfaz con el Filesystem	x		x		
12	Implementación de Filesystem	x		x		
13	Organización de Disco	x		x		
14	Parcial 1ra Fecha					x
15	Parcial 2ra Fecha					x
16	Presentación de trabajos prácticos					x

