

1. Objetivos

Los objetivos de la asignatura de Sistemas Operativos pueden agruparse de la siguiente manera:

a- De la cátedra

Orientar y coordinar el Aprendizaje por parte del alumno que curse la materia en los conceptos fundamentales de la arquitectura y diseño de los Sistemas Operativos (SOp) vigentes en el mercado.

Introducirlo en el desarrollo de Programas y Aplicación de Utilidades que conforman los Sistemas Operativos, en los ambientes de trabajo del futuro profesional en Informática.

b- Del alumno

Conformar un aprendizaje teórico-práctico, sobre los Sistemas Operativos en uso, llevado de una intensa actividad de cursado e investigación que volcará luego en la confección y aprobación de un conjunto de trabajos prácticos y evaluaciones parciales teórico-prácticas.

Lograr una concepción global y un enfoque selectivo, para realizar soluciones algorítmicas, de desarrollo de programas y aplicación de métodos de solución, de los diferentes problemas que se presentan entre la interacción del Hardware y los SOp, con la orientación de solución de problemas de instalación, configuración y Explotación de Sistemas Informáticos y Computacionales.

c- Según Contenidos Mínimos

Deben ser cubiertos y desarrollados los siguientes CMs:

Sistemas Operativos, concepto, evolución, estructura y servicios que brindan. Inicialización y configuración de un Sistema Operativo. Virtualización. Concepto de proceso, comunicación entre procesos, problemas de sincronización y comunicación, uso de IPC, planificación de procesos. Sistema de archivos, protección. Administración de memoria, memoria virtual, su administración. Administración de entrada/salida. Seguridad del Sistema Operativo.

2. Contenidos

La Cátedra de Sistemas Operativos se encuentra orientada a que los Alumnos desarrollen las siguientes bases conceptuales y prácticas.

Repaso de la estructura general y funcionalidad de un Sistema Operativo. Procesos como unidad de ejecución. Thread o hilos de Ejecución. (Unidad 1)

Problemas y métodos de Comunicación entre módulos de ejecución. Estructuras de Programación aplicadas a la intercomunicación (Unidades 2 y 3)

Estructura, operación y diseño de un Sistema de Archivos.(Unidad 4)

Práctica y desarrollo de aplicaciones que conforman los diversos Administradores de un Sistema Operativo. (Unidades 5 y 6)

Seguridad en los Sistemas Operativos, tanto en Clientes como en Servidores. (Unidad 7)

Unidad I: Introducción y planificación de procesadores

1. Arquitectura de los Sistemas Operativos: Concepto, evolución, estructura y servicios que brindan.
2. Ambientes de trabajo. Centralizado, Descentralizado y Distribuido. Inicialización y configuración de Sistemas Operativos. Virtualización. Sistemas Operativos de Red.
3. Introducción a los problemas del procesador: modelo de estados de los procesos. Planificación de trabajos. Ordenamiento de los procesos. Orden de ejecución de procesos, trabajos y thread. Colas y prioridades.
4. Planificación de Procesos
Conceptos básicos. Introducción. Planificación en Largo, Mediano y Corto Plazo. Planificación de los Procesos: No-preemptive y Preemptive scheduling. Algoritmos de Planificación No-Preemptive: FCFS, SJF y Prioridad. Comentarios. Algoritmos de Planificación Preemptive: Round-Robin, Múltiples colas fijas, Múltiples colas dinámicas. Comentarios.
5. Gestión de Procesos en Sistemas Distribuidos. Planificación de Procesos en ambientes multiprocesador distribuidos.
6. Gestión de Procesos en Unix - Linux.
7. Entorno de desarrollo de Procesos:
Conceptos y definiciones.
Atributos de un proceso.
Manipulación de Procesos.
Ejemplos de desarrollo.

Unidad II: Sincronización e IPC

1. Concurrencia. Principios Generales.
2. Exclusión Mútua. Soluciones por Sw y Hw. Abrazo Mortal (Deadlocks, Interbloqueo o bloqueo mutuo): Condiciones. Inanición. Métodos de Solución.
3. IPCs: Pipes & FIFO, Espacios de Memoria Compartida, Semáforos, Monitores, Pasaje de Mensajes y Sockets. Problema de Productor-Consumidor.
4. Entorno de Desarrollo:
Proceso de compilación. Compilador GNU (gcc).
Control del proceso de compilación, make de GNU.
Administración de errores.
Ejemplos de desarrollo.
Solución a problemas de sincronización y de uso de IPC.

Unidad III: Señales y Llamadas del Sistema

1. Conceptos sobre señales e Interrupciones.
2. Envío de señales. Intercepción de señales.
3. Llamadas al sistema. Utilización de llamadas al sistema.
4. Entorno de Desarrollo. Ejemplos aplicación de señales e interrupciones.

Unidad IV: Sistema de Archivos

1. Introducción. Organización y acceso a archivos: Pilas, archivos secuenciales, archivos secuenciales indexados, archivos indexados y archivos directos o de dispersión. Organización de directorios: Contenidos, estructura y designación. Protección.
2. Compartición de archivos: Derechos de acceso, acceso simultáneos, agrupación de registros.
3. Gestión del almacenamiento secundario: Asignación de archivos, gestión del espacio libre y fiabilidad.
4. Gestión de archivos en Unix/Linux: i-nodo y asignación de archivos.
5. Entorno de Desarrollo. Ejemplos aplicación administración de Archivos..

Unidad V: Administración de entrada-salida.

1. Introducción. Funciones generales.
2. Necesidad de la administración de periféricos. Controladores y canales. Implementación de controles.
3. Esquema de asignación de periféricos: exclusiva, compartida o virtual, automática o por operador. Consideraciones a tener en cuenta para compartir discos entre varios SOP
4. Principios del Hardware entrada-salida: device controlers y drivers. DMA y dispositivos de entrada-salida. Discos. Terminales. Cloks. Handlers de dispositivos y de IRQs. Concepto de RAID.
5. Concepto de método de acceso: acceso como interfase estándar entre programas y hardware, funciones de los métodos de acceso, clases de métodos de acceso, invocación a los métodos de acceso, ejemplos.
6. Concepto de Buffer de datos. Concepto de bloqueo y desbloqueo de registros. Concepto de programa de Canal.
7. Funciones de supervisor de entrada-salida. Administración de canales y periféricos: algoritmo de selección, encolamiento de requerimiento de entrada-salida (por periférico, por canal, por red).
8. Ordenamiento de la entrada-salida: estado y asignación.

Unidad VI: Administración de Memoria

1. Conceptos de la Administración de Memoria. Concepto de rutinas reusables y reentrantes. Esquemas de administración de memoria: monoprogramación y multiprogramación.
2. Protección de memoria: necesidad de importancia de protección de memoria en los Sistemas Operativos multiprogramables, esquemas de protección (registros base y límite, clave de protección).
3. Administración de Memoria por Particiones. Fijas, Variable, Sistema de Colegas. Problemas de Fragmentación.
4. Administración de Memoria Virtual. El swapping: concepto. swap-in, swap-out, el tiempo del multiplexado de los recursos (trashing).
5. La paginación: concepto de página y cuadro de página. Las tablas y el direccionamiento. Funciones implementadas por el Hardware y por el SOP. Fragmentación interna. Distintos algoritmos de reemplazo de página. .
6. Segmentación Combinación de segmentación y administración por demanda. Ventajas y desventajas de los sistemas de segmentación.
7. Entorno de Desarrollo. Ejemplos aplicación administración de Memoria.

Unidad VII: Seguridad en Sistemas Operativos

1. Protección: Introducción al problema, diferencia entre protección y seguridad. Objetivos de la protección, dominios, matriz de accesos, lista de controles de accesos, revocación de derechos de acceso. Sistemas basados en capacidades y objetos. Protección basada en lenguajes.
2. Seguridad: Introducción al problema. El problema de la seguridad. La autenticación. Seguridad externa: física y administración. Seguridad interna: de operación, supervisión de amenazas. Protección por contraseñas. Controles de acceso.
3. Seguridad por Hardware, por Software. Hilos de programas (program threads). Hilos de sistema (system free). Monitoreo de los hilos (threads monitoring). Encartación. Virus, troyanos, gusanos.

3. BIBLIOGRAFIA

3.1. Principal - Primaria

Programación en Linux, con ejemplos. Kurt Wall. QUE, Prentice Hall. Madrid. 2000.
Sistemas Operativos, principios de diseño e interioridades. Stallings William. 5ta Edición (Esp). Prentice Hall. Madrid. 2006.
Sistemas Operativos Modernos. Andrew. S. Tanenbaum. Prentice-Hall. Interamericana S.A. Madrid, 2009.

3.2. Consulta - Secundaria

El Libro de Linux. Syed M. Sarwar, Robert Koretsky y Syed. A. Sarwar. Ed. Addison Wesley. 2007. España.
Sistemas Operativos. Gary Nutt. 3er. Edición. Ed. Pearson Educación. 2004.
Sistemas operativos. Conceptos fundamentales. Silberschatz-Peterson-Galvin. 3ra. Edición Addison Wesley. 1994.
Sistemas Operativos, un enfoque basado en conceptos. D.M. Dhamdhere. Ed. Mc Graw Hill. Mexico DF. 2008.
Creación de Distribuciones Linux. Personalización para cada necesidad. Claudio Peña Millahual. Edit. MP. Manuales de Users. 2009.

3.3. Lectura de Referencia

Administración de Sistemas Linux. Guía Avanzada. M.Carling, Stephen Degler y James Dennis. Prentice Hall, 2000.
Operating Systems, 3th edition. William Stallings. Prentice Hall, 2001.
Sistemas Operativos, Diseño e implementación. Andrew Tanenbaum. Prentice Hall, 1994 .
Sistemas Operativos Modernos. Andrew Tanenbaum, Prentice Hall, 1996.
Cátedra de Sistemas Operativos, Apuntes. Ing. Netzeel. UTN. 1997.
El Libro de Unix. Syed M. Sarwar, Robert Koretsky y Syed A. Sarwar. Ed. Addison Wesley, 2004. España.
UNIX Sistema V Versión 4. Rosen, Rozinsky y Farber. McGraw Hill. Edic. 2002.
LINUX, Edición especial. Tackett Jack, Gunter David y Brown Lance. Ed. Prentice Hall. 1998.
UNIX para impacientes. Abrahams. Addison Wesley Iberoamericana.
La Arquitectura de Windows NT a fondo. Solomon. Microsoft. 2000

Manual de Seguridad en Windows NT. Sheldon. Microsoft. 2001

Linux firewalls. Ziegler. New Riders Publishing. 2002

Manual de Actualización y reparación de PCs, 12 edición. Scott Mueller. Que, Prentice Hall, 2001.

Sistemas Operativos una Visión Aplicada. Jesús Carretero Pérez, Pedro de Miguel Anasagasti, Félix García Carballeira y Fernando Pérez Costoya. 1er Edición. Editorial McGraw-Hill. 2001.

4. METODOLOGIA DE LA ENSEÑANZA

El proceso educativo parte de la descripción y análisis de situaciones y problemas, introduciendo métodos útiles para resolverlos y aplicarlos utilizando herramientas computacionales que aprenden a utilizar los estudiantes.

El enfoque es un Taller que se dicta en un Laboratorio de Informática. La actividad comienza con el desarrollo de los contenidos conceptuales con ayuda de elementos audiovisuales y la descripción de ejemplos sencillos. Sobre la base de los problemas se presentan los modelos que los representan y desarrollan los métodos de resolución.

Una vez captado el método por parte de los estudiantes, se utilizan Sistemas Operativos usados en el mercado de trabajo para resolver casos de estudio, realizar prácticas supervisadas, por parte de los estudiantes. A dicho trabajo se agrega la discusión sobre material obtenido de Internet y una relación frecuente mediante correo electrónico con el profesor para efectuar consultas y recibir correcciones.

La Cátedra ha desarrollado problemas y casos de estudio para los distintos temas, los que sirven para ejemplificar los métodos de operación y diseño que se utilizan en los Sistemas Operativos.

El alumno tendrá acceso a un conjunto de Actividades que le permitirá conformar su entorno de Aprendizaje, los cuales se desarrollarán en los siguientes lugares:

- a. Actividades de Enseñanza en el Aula. Clases Grupales de tipo Teórico.
- b. Actividades de Práctica en Laboratorios de Computadoras, que le permitirán familiarizarse con el ambiente de trabajo, y desde allí construir en la operación su Aprendizaje.
- c. Actividades de Investigación aplicada en los Trabajos Prácticos, de tipo Grupal. Cada Investigación deberá concluir con la correspondiente **presentación de la documentación** por mail y con el correspondiente medio de almacenamiento (disquette – cd – dvd). En todos los casos la correcta funcionalidad es la base de la corrección y aprobación de la asignatura.

Esto permitirá al alumno:

Adquirir vocabulario técnico-informático y utilizarlo con precisión

Conocer en forma amplia y general el funcionamiento de las partes de un SOP.

Evaluar, a nivel de implementación, cualquier SOP. sobre equipos existentes en plaza.

Desarrollar en el alumno el interés por la investigación; utilizando publicaciones, libros y sistemas reales que sean propuestos por el Profesor.

Ayudar a desarrollar en el alumno la actitud de detector de posibles soluciones de problemas, realizando una adecuada ejercitación práctica.

Iniciarse en las actividades de trabajo que conlleva el desarrollo colaborativo con el propio grupo o con grupos más extensos, a nivel local y remoto.

Detalle de Actividades prácticas

Lo anterior será posible materializarlo con trabajos de investigación, desarrollo de aplicaciones y evaluaciones de tipo Individual y Grupal. Estas pueden listarse de la siguiente forma:

Formación experimental (P1)

Se resuelven problemas que ilustran la teoría mediante ejemplos que se plantean en el pizarrón y luego se resuelven mediante las herramientas del Sistema Operativo elegido. Los problemas ofrecen dificultades crecientes y en algunos casos son versiones simplificadas de problemáticas reales. Trabajamos con Sistema Operativos de la familia Unix-Linux y Windows.

Problemas abiertos de ingeniería (P2)

Son problemas que corresponden a situaciones reales o hipotéticas cuya solución requiera la aplicación de los conocimientos de las ciencias básicas y de las tecnologías utilizadas en Sistemas Operativos. En general no tienen un planteo matemático único, sino que dependerá de los requerimientos de los que toman las decisiones y los límites que pueden plantearse a la complejidad. Las conclusiones deben presentarse en informes grupales, que deben resultar útiles a quien tome decisiones en el diseño e implementación de los Sistemas Operativos.

Prácticas de proyecto y diseño de sistemas informáticos(P3)

Se entiende por tales a las actividades que empleando ciencias básicas y de la ingeniería llevan al desarrollo de un sistema, componente o proceso, satisfaciendo una determinada necesidad y optimizando el uso de los recursos disponibles. Corresponde a los casos más complejos planteados, donde los alumnos deben relacionar conceptos de matemática, economía, sistemas y toma de decisiones. Las conclusiones deben presentarse en informes grupales, que deben resultar útiles a quien tome decisiones.

Instrucción Supervisada de Formación Práctica (P4)

Se entiende por tales a las actividades que empleando diversas herramientas de software y hardware permiten conformar un conocimiento práctico, aplicable al ámbito profesional. Son actividades grupales y se realizan en forma concentrada en los Laboratorios, guiados por el Docente.

Conformación de los Trabajos Prácticos y Ensayos a realizar por los Alumnos a lo largo de su cursado.

Unidad Temática	TP – Aplicativo	Comentarios
I. Estructuras de Ejecución	TP 1. Utilidades de Desarrollo – Instalación y Configuración de SOp.	Instalación y Comandos de Linux.
	TP 2. Procesos y Estructuras de ejecución.	Uso de gcc y resolución de problemas de planificación de procesos.
II. Sincronización e IPC	TP 3. Sincronización e IPC.	Planificación y Solución a problemas de diseño.
III. Señales y Llamadas al Sistema	TP 4. Señales y Llamadas de Sistema.	Programación de Señales.
IV: Sistemas de Archivos	TP 5. Sistema de Archivos.	Solución a problemas de administración de archivos
V: Administración de Entrada/Salida	TP 6. Sistemas de I/O y Memoria.	Resolución de problemas de memoria e I/O, usando métodos pre-establecidos
VI: Administración de Memoria		
VII: Seguridad en Sistemas Operativos	TP 7. Seguridad en SOp.	Aplicación de problemas de transferencia de información.

Los Alumnos además deberán desarrollar una serie de Ensayos, con formato predefinido, en los cuales se explicitarán sobre temas concernientes a los vistos en Aula, y deberán ser entregados, con una frecuencia de 15 días. Los temas propuestos son:

Es la virtualización una salida práctica y de bajo costo para compensación de Servidores?

El Software Libre puede ser una opción hoy para el segmento bancario y financiero.?

Pueden desarrollarse arquitecturas de SOps sin administradores ni drivers?

Para trabajar con las herramientas de SOps que prefiere, Shell carácter o Shell gráfico, porque?

Porque las empresas del medio argentino prefieren BSD sobre Linux?

Que es más complejo, programar un juego, o un administrador de scheduler?

Todos los ensayos deben declarar bibliografía y otras fuentes, además no pueden superar el 30% del escrito, lo textual de terceros. Aquí se puede plantear del criterio restante; presentación de temas específicos por alumno.

5. CRITERIOS DE EVALUACION

La evaluación de los alumnos se realiza a través de Trabajos Prácticos (TPs), participación en clases, evaluaciones parciales y el Examen Final.

En los TPs: los alumnos deberán poner en juego las competencias desarrolladas y los conocimientos adquiridos mediante la resolución de problemas. Se tomarán en cuenta el contenido, el cumplimiento de objetivo y consignas y la calidad de la presentación (prolijidad, ortografía, comunicación).

En la participación en clase: Los alumnos serán evaluados en forma permanente a través de la calidad y oportunidad de sus intervenciones.

En los Parciales: la evaluación parcial tiene como objetivo corroborar el aprendizaje realizado por los alumnos durante el curso y su evolución. Se verificará el nivel de cumplimiento de los objetivos pedagógicos del curso.

En el Examen Final: La evaluación final estará basada sobre la examinación del conocimientos vistos en la materia y resolver problemas reales que permitan poner en evidencia la integración de conocimientos. Se verificará la capacidad de los alumnos en la utilización de los conceptos fundamentales de la asignatura para la organización de su trabajo, así como el nivel de análisis desarrollado y la calidad de la solución propuesta.

5.2 Requisitos para la aprobación

Aprobación del cursado de la asignatura. Para aprobar es necesario cumplir con:

Asistencia mínima del 50%

Aprobación del examen parcial con nota igual o superior a cuatro puntos:

Los parciales deben rendirse en las fechas estipuladas por la Facultad, según cronograma general de la Universidad.

En el caso de que el alumno desaprobe el examen parcial cuenta con una instancia de recuperación.

El desaprobar o no asistir a la recuperación (teniendo el parcial desaprobado) tiene como consecuencia desaprobar el curso de la materia.

Aprobación de los Trabajos prácticos con nota igual o superior a cuatro puntos:

En el caso de esta materia la nota final de los trabajos prácticos se calcula como una nota promedio de los trabajos requeridos que equivale al 75% del número de TPs obligatorios.

Aprobación de la asignatura. Para aprobar la materia es necesario aprobar el cursado y el Examen Final

Para aquellos alumnos que no alcanzaran el 75% de asistencias deberán rendir un Examen Final Escrito y luego un Examen Final Oral.

Para los alumnos que alcancen o superen el 75% el Examen Final será sólo de tipo Oral.