

 <b>UNIVERSIDAD NACIONAL DE CÓRDOBA</b> Facultad de Ciencias Exactas, Físicas y Naturales República Argentina	Programa de:  <h2 style="text-align: center;">Sistemas Operativos II</h2>  Código: 2637																
Carrera: <i>Ingeniería en Computación</i> Escuela: <i>Ingeniería Electrónica y Computación.</i> Departamento: <i>Computación.</i>	<table border="0" style="width: 100%;"> <tr> <td>Plan:</td> <td>285-05</td> <td>Puntos:</td> <td>4</td> </tr> <tr> <td>Carga Horaria:</td> <td>96</td> <td>Hs. Semanales:</td> <td>6</td> </tr> <tr> <td>Semestre:</td> <td>Noveno</td> <td>Año:</td> <td>Quinto</td> </tr> <tr> <td>Carácter:</td> <td colspan="3"><i>Obligatoria</i></td> </tr> </table>	Plan:	285-05	Puntos:	4	Carga Horaria:	96	Hs. Semanales:	6	Semestre:	Noveno	Año:	Quinto	Carácter:	<i>Obligatoria</i>		
Plan:	285-05	Puntos:	4														
Carga Horaria:	96	Hs. Semanales:	6														
Semestre:	Noveno	Año:	Quinto														
Carácter:	<i>Obligatoria</i>																
<b>Objetivos:</b> <ul style="list-style-type: none"> <li>• <i>Comprender el objetivo de la planificación de los sistemas mono y multiprocesador. Diferenciar entre sistemas de propósito general y específicos para aplicaciones determinadas. Influencia de estos elementos en planificación.</i></li> <li>• <i>Comprender el concepto de procesos distribuidos, su gestión y el efecto sobre la seguridad de un sistema.</i></li> <li>• <i>Introducir al educando en los sistemas de tiempo real, su implementación, programación de actividades puras de tiempo real y su relación con procesos estándar. Observar y estudiar la comunicación entre procesos de fase crítica y procesos estándar.</i></li> </ul>																	
<b>Programa Sintético:</b> <ol style="list-style-type: none"> <li>1. <i>Conceptos de planificación mono y multiprocesador.</i></li> <li>2. <i>Procesos distribuidos y seguridad.</i></li> <li>3. <i>Mecanismos cliente/servidor y clusters. Gestión.</i></li> <li>4. <i>Ejecución de procesos y comunicación.</i></li> <li>5. <i>Sistemas de tiempo real.</i></li> <li>6. <i>Implementaciones de Tiempo Real</i></li> <li>7. <i>Sistemas embebidos.</i></li> </ol>																	
Programa Analítico: de foja 4 a foja 5.																	
Programa Combinado de Examen (si corresponde): de foja    a foja    .																	
Bibliografía: de foja 3 a foja 3.																	
Correlativas Obligatorias: <i>Ingeniería de Software</i> <i>Sistemas Operativos I</i>																	
Correlativas Aconsejadas:																	
Rige: 2005																	
<table border="0" style="width: 100%;"> <tr> <td style="width: 50%;">Aprobado HCD, Res.:</td> <td style="width: 50%;">Modificado / Anulado /Sust. HCD Res.:</td> </tr> <tr> <td>Fecha:</td> <td>Fecha:</td> </tr> </table> <p>El Secretario Académico de la Facultad de Ciencias Exactas, Físicas y Naturales (UNC) certifica que el programa está aprobado por el (los) número(s) y fecha(s) que anteceden. Córdoba,    /    /    .</p>		Aprobado HCD, Res.:	Modificado / Anulado /Sust. HCD Res.:	Fecha:	Fecha:												
Aprobado HCD, Res.:	Modificado / Anulado /Sust. HCD Res.:																
Fecha:	Fecha:																
Carece de validez sin la certificación de la Decretaría Académica:																	

## PROGRAMA ANALITICO

### LINEAMIENTOS GENERALES

Sistemas Operativos II es una actividad curricular que pertenece al noveno semestre de la carrera de Ingeniería en Computación.

En los últimos tiempos se ha experimentado una gran evolución de los sistemas operativos, especialmente a partir de la aparición y afianzamiento de la programación open source. La posibilidad de que los códigos sean accesibles a cualquiera ha llevado a que las aplicaciones y funciones en las que se los usa, excedan ampliamente el objetivo inicial.

Entre estas características podemos destacar:

- Modificaciones a los sistemas estándar para trabajar como tiempo real.
- Minimización de componentes para ser insertados en sistemas embebidos.
- Herramientas de todo tipo para la evolución de los dos puntos anteriores..

Esto ha hecho que los sistemas operativos ingresen a lugares a los que les estaba vedado, por espacio, capacidades de procesador, etc.

Por otra parte, pudiendo acceder a los componentes, se ha reemplazado el núcleo original, por versiones deterministas que permiten el trabajo en tiempo real, dejando al sistema operativo original como la tarea de menor prioridad entre las que realiza el nuevo planificador.

A partir de esta concepción, la materia enfoca dos líneas claramente definidas:

- **Diseño de programas de tiempo real:** Ser capaz, a partir de especificaciones, de diseñar e implementar un sistema de tiempo real estricto.
  - Reconocer que un sistema instalado está preparado para tiempo real.
  - Ejercitar pruebas de benchmarking para certificar que el sistema responderá a un conjunto de requerimientos.
  - Aprender las actividades mas importantes del proceso de diseño e implementación de sistemas de tiempo real
- **Pruebas de Funcionamiento:** Ser capaz de diseñar y realizar pruebas en sistemas que funcionen en tiempo real.
  - Establecer un conjunto de parámetros de prueba para ensayo del sistema
  - Ser capaz de modificar parámetros que permitan la adaptación de un sistema de tiempo real a una necesidad específica.
  - Ser capaz de trasladar un sistema de tiempo real a un sistema embebido, comprobando que el funcionamiento concuerda con los valores obtenidos en emulación.

La enseñanza está dirigida a comprender el funcionamiento de los sistemas de tiempo real, abordando previamente los conceptos que serán necesarios para el tratamiento del tema y enfatizando el diseño de sistemas de tiempo real a partir de un conjunto de restricciones (especificaciones).

También se enfoca el trabajo de sistemas operativos en sistemas embebidos, con aplicación a controladores industriales.

## **METODOLOGIA DE ENSEÑANZA**

Las clases impartidas son teóricas, prácticas y de laboratorio.

Las actividades teóricas se realizan a través de exposiciones del docente orientadas a desarrollar en los alumnos la capacidad de implementar sistemas de tiempo real.

Durante el desarrollo de los Trabajos Prácticos se realizan actividades que le permiten al estudiante poner en práctica las habilidades y verificar los criterios diseño y verificación.

Por otra parte en las clases de Laboratorio el alumno verifica, a través de la implementación, el funcionamiento de sistemas y realizará los casos de pruebas.

## **EVALUACION**

### **Condiciones para la promoción de la materia**

1. Tener aprobadas las materias correlativas.
2. Asistir al 80% de las clases teóricas y prácticas.
3. Rendir y aprobar los dos parciales con 70% , el alumno podrá recuperar un parcial. Para los que aprueban con un 50% (mínimo necesario) se rendirá con coloquio integrador.
4. Presentar y aprobar las implementaciones que se exijan durante el desarrollo de los trabajos prácticos y un trabajo final.
5. Aprobar los trabajos de Laboratorio.

Los alumnos que cumplan las exigencias referidas serán considerados promocionados.

Los alumnos que hayan aprobado los parciales con un 50% y no hayan realizado o aprobado el coloquio integrador serán considerados regulares. El trabajo final lo deben presentar para el examen de la materia.

El resto será considerado libre.

## CONTENIDOS TEMATICOS

### Unidad 1. Planificación monoprocesador

1. Tipos de planificación
2. Algoritmos de planificación
3. Planificación de Unix tradicional
4. Resumen.

### Unidad 2. Planificación multiprocesador y en tiempo real

1. Planificación de multiprocesadores
2. Planificación en tiempo real
3. Planificación Linux
4. Planificación Unix SVR4
5. Planificación Windows 2000
6. Resumen.

### Unidad 3. Proceso distribuido, cliente/servidor. Clusters

1. Computación cliente/servidor.
2. Proceso distribuido mediante paso de mensajes,
3. Llamadas a procesos remotos.
4. Clusters.
5. Windows 2000 cluster server
6. Resumen.

### Unidad 4. Gestión de procesos distribuidos

1. Migración de procesos
2. Estado global distribuido
3. Gestión distribuida de procesos.
4. Exclusión mutua.
5. Interbloqueo distribuido
6. Resumen

### Unidad 5. Procesos

1. Procesos y señales.
2. Terminación.
3. Intercambio de datos.
4. Atributos.
5. Valores de contorno.
6. Cancelación.
7. Handlers.
8. Sincronización y secciones críticas.
9. Exclusión mutua.
10. Llamadas de sistema.
11. Manejo de tiempos.
12. Temporización.

**Unidad 6. Comunicación entre procesos**

1. Introducción
2. Procesos padre e hijo
3. Pipes
4. Named Pipes (FIFOs)
5. Colas de mensajes.
6. Memoria compartida.
7. Semáforos.
8. Sockets.

**Capítulo 7. Sistemas embebidos**

1. Concepto de sistema embebido
2. Restricciones de los sistemas embebidos
3. Sistemas operativos para sistemas embebidos
4. Adaptación de sistemas operativos a sistemas embebidos y modificaciones del kernel en sistemas abiertos.
5. Ambiente de desarrollo y compilación cruzada.
6. Depuración de programas para sistemas embebidos

**Capítulo 8. Sistemas de tiempo real**

1. Tiempo de respuesta.
2. Definiciones.
3. Planificación
4. Performance
5. Diseño de sistemas de tiempo real.
6. Fiabilidad y tolerancia a fallos.
7. Lenguajes de programación para sistemas de tiempo real.

**Capítulo 9. Implementaciones de tiempo real**

1. Aproximaciones de Linux al tiempo real.
2. Versiones disponibles.
3. Real Time subkernel.
4. Ejemplos.
5. Resumen.

## LISTADO DE ACTIVIDADES PRÁCTICAS Y/O DE LABORATORIO

### Actividades Prácticas

#### 1. Práctica con planificación monoprocesador

Gráfico de tiempos de ejecución de tareas.  
Ejecución de tabla comparativa de performance

#### 2. Práctica con multiprocesador y tiempo real

Verificación de cumplimiento de plazos en tiempo real.

#### 3. Práctica de comunicación entre procesos

Resolución de problemas de comunicación básico.  
Problemas de comunicación avanzada.

#### 4. Comunicación de procesos en computación paralela

MPI

#### 5. Instalación de soft y verificación de un sistema embebido

Tareas preparatorias. Análisis de memorias en solo lectura.  
Ejecución de tareas de sensado y control a partir de un sistema embebido.

#### 6. Instalación y benchmarking de un sistema de tiempo real

Ensayo y verificación de cumplimiento de plazos con tiempo real rígido.

### Actividades de Laboratorio

Cálculo de secuenciamiento de tareas en sistemas monoprocesador.  
Cálculo de secuenciamiento de tareas en sistemas multiprocesador.  
Cálculo de secuenciamiento de tareas en tiempo real.  
Práctica de comunicación entre procesos.  
Diseño de clusters. Cálculos de performance.  
Diseño de sistemas de tiempo real rígido

**DISTRIBUCION DE LA CARGA HORARIA**

<b>ACTIVIDAD</b>	<b>HORAS</b>
TEÓRICA	40
FORMACIÓN PRACTICA:	
○ FORMACIÓN EXPERIMENTAL	11
○ RESOLUCIÓN DE PROBLEMAS	20
○ ACTIVIDADES DE PROYECTO Y DISEÑO	25
<b>TOTAL DE LA CARGA HORARIA</b>	<b>96</b>

**DEDICADAS POR EL ALUMNO FUERA DE CLASE**

<b>ACTIVIDAD</b>	<b>HORAS</b>
PREPARACION TEÓRICA	37
PREPARACION PRACTICA	
○ EXPERIMENTAL DE LABORATORIO	13
○ EXPERIMENTAL DE CAMPO	
○ RESOLUCIÓN DE PROBLEMAS	30
○ PROYECTO Y DISEÑO	35
<b>TOTAL DE LA CARGA HORARIA</b>	<b>115</b>

## BIBLIOGRAFIA

### Principal

1. Sistemas Operativos. Williams Stallings. Prentice Hall.
2. Sistemas Operativos Modernos. Andrew Tanenbaum. Prentice Hall.
3. Fundamentos de Sistemas Operativos. Silberschatz, Galvin y Gagne. Mc Graw Hill.
4. Programación en Linux con ejemplos. Kurt Wall. Prentice Hall.
5. Advanced Linux Programming. Kurt Wall. Prentice Hall,
6. Sistemas de Tiempo Real y Lenguajes de Programación. Alan Burns y Andy Wellings. Addison Wesley.

### Complementaria

1. Understanding the Linux Kernel, 2nd Edition. Daniel P. Bovet, Marco Cesati. O'Reilly
2. Programming Embedded Systems in C and C++. Michael Barr. O'Reilly.
3. Linux Device Drivers. Alessandro Rubinni. O'Reilly