

## 1. COURSE

CS2S1. Operating systems (Mandatory)

## 2. GENERAL INFORMATION

2.1 Course	:	CS2S1. Operating systems
2.2 Semester	:	4 <sup>to</sup> Semestre.
2.3 Credits	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duration of the period	:	16 weeks
2.6 Type of course	:	Mandatory
2.7 Learning modality	:	Face to face
2.8 Prerequisites	:	CS221. Computer Systems Architecture. (3 <sup>rd</sup> Sem) CS221. Computer Systems Architecture. (3 <sup>rd</sup> Sem)

## 3. PROFESSORS

Meetings after coordination with the professor

## 4. INTRODUCTION TO THE COURSE

An Operating System (OS) manages the computing resources to complete the execution of multiple applications and their associated processes. This course teaches the design of modern operating systems; and introduces their fundamental concepts covering multiple-program execution, scheduling, memory management, file systems, and security. Also, the course includes programming activities on a minimal operating system to solve problems and extend its functionality. Notice that these activities require much time to complete. However, working on them provides valuable insight into operating systems.

## 5. GOALS

- Study the design of modern operating systems.
- Provide a practical experience by designing and implementing a minimal operating system.

## 6. COMPETENCES

- 1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. (**Assessment**)
- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (**Familiarity**)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (**Usage**)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (**Usage**)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (**Assessment**)

## 7. TOPICS

Unit 1: Visión general de Sistemas Operativos (3)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Papel y el propósito del sistema operativo.</li> <li>• Funcionalidad de un sistema operativo típico.</li> <li>• Los mecanismos de apoyo modelos cliente-servidor.</li> <li>• Cuestiones de diseño (eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad)</li> <li>• Influencias de seguridad, creación de redes, multimedia, sistemas de ventanas.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar los objetivos y funciones de un sistema operativo moderno [Familiarity]</li> <li>• Analizar las ventajas y desventajas inherentes en el diseño de un sistema operativo [Assessment]</li> <li>• Describir las funciones de un sistema operativo contemporáneo respecto a conveniencia, eficiencia, y su habilidad para evolucionar [Familiarity]</li> <li>• Discutir acerca de sistemas operativos cliente-servidor, en red, distribuidos y cómo se diferencian de los sistemas operativos de un solo usuario [Familiarity]</li> <li>• Identificar amenazas potenciales a sistemas operativos y las características del diseño de seguridad para protegerse de ellos [Familiarity]</li> </ul>
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 2: Principios de Sistemas Operativos (6)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Estructuración de Sistemas Operativos (monolítico, capas, modular, los modelos micro-kernel)</li> <li>• Abstracciones, procesos y recursos.</li> <li>• Los conceptos de interfaces de programa de aplicación (API)</li> <li>• La evolución de las técnicas de hardware / software y las necesidades de aplicación</li> <li>• Organización de dispositivos.</li> <li>• Interrupciones: métodos e implementaciones.</li> <li>• Concepto de estado de usuario / sistema y la protección, la transición al modo kernel.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar el concepto de una capa lógica [Familiarity]</li> <li>• Explicar los beneficios de construir capas abstractas en forma jerárquica [Familiarity]</li> <li>• Describir el valor de la API y <i>middleware</i> [Familiarity]</li> <li>• Describir como los recursos computacionales son usados por aplicaciones de software y administradas por el software del sistema [Familiarity]</li> <li>• Contrastar el modo <i>kernel</i> y modo usuario en un sistema operativo [Assessment]</li> <li>• Discutir las ventajas y desventajas del uso de procesamiento con interrupciones [Familiarity]</li> <li>• Explicar el uso de una lista de dispositivos y el controlador de colas de entrada y salida [Familiarity]</li> </ul>
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 3: Concurrency (9)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Diagramas de estado.</li> <li>• Estructuras (lista preparada, bloques de control de procesos, y así sucesivamente)</li> <li>• <i>Dispatching</i> y cambio de contexto.</li> <li>• El papel de las interrupciones.</li> <li>• Gestionar el acceso a los objetos del sistema operativo de forma atómica.</li> <li>• La implementación de primitivas de sincronización.</li> <li>• Problemas de multiprocesador (spin-locks, reentrada)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir la necesidad de concurrencia en el marco de un sistema operativo [Familiarity]</li> <li>• Demostrar los potenciales problemas de tiempo de ejecución derivados de la operación simultánea de muchas tareas diferentes [Usage]</li> <li>• Resumir el rango de mecanismos que pueden ser usados a nivel del sistema operativo para realizar sistemas concurrentes y describir los beneficios de cada uno [Familiarity]</li> <li>• Explicar los diferentes estados por los que una tarea debe pasar y las estructuras de datos necesarias para el manejo de varias tareas [Familiarity]</li> <li>• Resumir las técnicas para lograr sincronización en un sistema operativo (por ejemplo, describir como implementar semáforos usando primitivas del sistema operativo.) [Familiarity]</li> <li>• Describir las razones para usar interrupciones, <i>dispatching</i>, y cambio de contexto para soportar concurrencia en un sistema operativo [Familiarity]</li> <li>• Crear diagramas de estado y transición para los problemas de dominios simples [Usage]</li> </ul>
<b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 4: Planificación y despacho (6)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• <i>Scheduling preemptive y non-preemptive.</i></li> <li>• <i>Scheduling</i> y políticas.</li> <li>• Procesos y subprocesos.</li> <li>• Plazos y cuestiones en tiempo real.</li> </ul>	<ul style="list-style-type: none"> <li>• Comparar y contrastar los algoritmos comunes que se utilizan tanto para <i>scheduling preemptive</i> y <i>preemptive</i> de tareas en los sistemas operativos, como la comparación de prioridad, el rendimiento, y los esquemas de distribución equitativa [Assessment]</li> <li>• Describir las relaciones entre los algoritmos de <i>scheduling</i> y dominios de aplicación [Familiarity]</li> <li>• Discutir los tipos de <i>scheduling</i> en procesadores en de corto, mediano, largo plazo y I/O [Familiarity]</li> <li>• Describir las diferencias entre procesos y <i>threads</i> [Familiarity]</li> <li>• Comparar y contrastar enfoques estáticos y dinámicos para <i>scheduling</i> en tiempo real [Assessment]</li> <li>• Discutir sobre la necesidad de <i>preemption</i> y <i>deadline scheduling</i> [Familiarity]</li> <li>• Identificar formas en que la lógica expresada en algoritmos de planificación son de aplicación a otros ámbitos, tales como I/O del disco, la programación de disco de red, programación de proyectos y problemas más allá de la computación [Familiarity]</li> </ul>
<b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 5: Manejo de memoria (6)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Revisión de la memoria física y hardware de gestión de memoria.</li> <li>• Conjuntos de trabajo y thrashing.</li> <li>• El almacenamiento en caché</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar la jerarquía de la memoria y <i>tradeoffs</i> de costo-rendimiento [Familiarity]</li> <li>• Resumir los principios de memoria virtual tal como se aplica para el almacenamiento en cache y paginación [Familiarity]</li> <li>• Evaluar las ventajas y desventajas en términos del tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y la velocidad del procesador [Assessment]</li> <li>• Describir las diferentes formas de asignar memoria a las tareas, citando las ventajas relativas de cada uno [Familiarity]</li> <li>• Describir el motivo y el uso de memoria caché (rendimiento y proximidad, dimensión diferente de como los caches complican el aislamiento y abstracción en VM) [Familiarity]</li> <li>• Estudiar los conceptos de <i>thrashing</i>, tanto en términos de las razones por las que se produce y las técnicas usadas para el reconocimiento y manejo del problema [Familiarity]</li> </ul>
<b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 6: Seguridad y protección (6)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Visión general de la seguridad del sistema .</li> <li>• Política / mecanismo de separación.</li> <li>• Métodos de seguridad y dispositivos.</li> <li>• Protección, control de acceso y autenticación.</li> <li>• Las copias de seguridad.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar la necesidad para la protección y seguridad en un sistema operativo [Familiarity]</li> <li>• Resumir las características y limitaciones de un sistema operativo usado para proporcionar protección y seguridad [Familiarity]</li> <li>• Explicar el mecanismo disponible en un OS para controlar los accesos a los recursos [Familiarity]</li> <li>• Realizar tareas de administración de sistemas sencillas de acuerdo a una política de seguridad, por ejemplo la creación de cuentas, el establecimiento de permisos, aplicación de parches y organización de backups regulares [Familiarity]</li> </ul>
<b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 7: Máquinas virtuales (6)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Tipos de virtualización (incluyendo Hardware / Software, OS, Servidor, Servicio, Red)</li> <li>• Paginación y la memoria virtual.</li> <li>• Sistemas de archivos virtuales.</li> <li>• Los Hypervisores.</li> <li>• Virtualización portátil; emulación vs aislamiento.</li> <li>• Costo de la virtualización.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar el concepto de memoria virtual y la forma cómo se realiza en hardware y software [Familiarity]</li> <li>• Diferenciar emulación y el aislamiento [Familiarity]</li> <li>• Evaluar virtualización de compensaciones [Assessment]</li> <li>• Discutir sobre hipervisores y la necesidad para ellos en conjunto con diferentes tipos de hipervisores [Familiarity]</li> </ul>
<b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 8: Manejo de dispositivos (6)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Características de los dispositivos serie y paralelo.</li> <li>• Haciendo de abstracción de dispositivos.</li> <li>• Estrategias de buffering.</li> <li>• Acceso directo a memoria.</li> <li>• La recuperación de fallos.</li> </ul>	<ul style="list-style-type: none"> <li>• Explique la diferencia clave entre dispositivos seriales y paralelos e identificar las condiciones en las cuales cada uno es apropiado [Familiarity]</li> <li>• Identificar la relación entre el hardware físico y los dispositivos virtuales mantenidos por el sistema operativo [Familiarity]</li> <li>• Explicar <i>buffering</i> y describir las estrategias para su aplicación [Familiarity]</li> <li>• Diferenciar los mecanismos utilizados en la interconexión de un rango de dispositivos (incluyendo dispositivos portátiles, redes, multimedia) a un ordenador y explicar las implicaciones de éstas para el diseño de un sistema operativo [Familiarity]</li> <li>• Describir las ventajas y desventajas de acceso directo a memoria y discutir las circunstancias en cuales se justifica su uso [Familiarity]</li> <li>• Identificar los requerimientos para recuperación de errores [Familiarity]</li> <li>• Implementar un controlador de dispositivo simple para una gama de posibles equipos [Usage]</li> </ul>
<b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

<b>Unit 9: Sistema de archivos (6)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Archivos: los datos, metadatos, operaciones, organización, amortiguadores, secuenciales, no secuencial.</li> <li>• Directorios: contenido y estructura.</li> <li>• Los sistemas de archivos: partición, montar / desmontar sistemas de archivos virtuales.</li> <li>• Técnicas estándar de implementación .</li> <li>• Archivos asignados en memoria.</li> <li>• Sistemas de archivos de propósito especial.</li> <li>• Naming, búsqueda, acceso, copias de seguridad.</li> <li>• La bitacora y los sistemas de archivos estructurados (log)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las decisiones que deben tomarse en el diseño de sistemas de archivos [Familiarity]</li> <li>• Comparar y contrastar los diferentes enfoques para la organización de archivos, el reconocimiento de las fortalezas y debilidades de cada uno. [Assessment]</li> <li>• Resumir cómo el desarrollo de hardware ha dado lugar a cambios en las prioridades para el diseño y la gestión de sistemas de archivos [Familiarity]</li> <li>• Resumir el uso de diarios y como los sistemas de archivos de registro estructurado mejora la tolerancia a fallos [Familiarity]</li> </ul>
<b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

<b>Unit 10: Sistemas <i>embedded</i> y de tiempo real (6)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Proceso y programación de tareas.</li> <li>• Los requisitos de gestión de memoria / disco en un entorno en tiempo real.</li> <li>• Los fracasos, los riesgos y la recuperación.</li> <li>• Preocupaciones especiales en sistemas de tiempo real.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir que hace a un sistema un sistema en tiempo real [Familiarity]</li> <li>• Explicar la presencia y describir las características de latencia en sistemas de tiempo real [Familiarity]</li> <li>• Resumir los problemas especiales que los sistemas en tiempo real presentan, incluyendo el riesgo, y cómo se tratan estos problemas [Familiarity]</li> </ul>
<b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

<b>Unit 11: Tolerancia a fallas (3)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Conceptos fundamentales: sistemas fiables y disponibles.</li> <li>• Redundancia espacial y temporal.</li> <li>• Los métodos utilizados para implementar la tolerancia a fallos.</li> <li>• Los ejemplos de los mecanismos del sistema operativo para la detección, recuperación, reinicio para implementar la tolerancia a fallos, el uso de estas técnicas para los servicios propios del sistema operativo.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar la importancia de los términos tolerancia a fallos, fiabilidad y disponibilidad [Familiarity]</li> <li>• Explicar en términos generales la gama de métodos para implementar la tolerancia a fallos en un sistema operativo [Familiarity]</li> <li>• Explicar cómo un sistema operativo puede continuar funcionando después de que ocurra una falla [Familiarity]</li> </ul>
<b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 12: Evaluación del desempeño de sistemas (3)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• ¿Por qué el rendimiento del sistema debe ser evaluado?</li> <li>• ¿Qué se va a evaluar?</li> <li>• Sistemas de políticas de rendimiento, por ejemplo, el almacenamiento en caché, de paginación, la programación, la gestión de memoria, y la seguridad.</li> <li>• Modelos de evaluación: analítica, simulación, o de implementación específico determinista.</li> <li>• Cómo recoger los datos de evaluación (perfiles y mecanismos de localización)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las medidas de rendimiento utilizados para determinar cómo el sistema funciona [Familiarity]</li> <li>• Explicar los principales modelos de evaluación utilizados para evaluar un sistema [Familiarity]</li> </ul>
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BASIC BIBLIOGRAPHY

- [AD14] Thomas Anderson and Michael Dahlin. *Operating Systems: Principles and Practice*. 2nd. Recursive Books, 2014. ISBN: 978-0985673529.
- [Avi12] Greg Gagne Avi Silberschatz Peter Baer Galvin. *Operating System Concepts, 9/E*. John Wiley & Sons, Inc., 2012. ISBN: 978-1-118-06333-0.
- [Sta05] William Stallings. *Operating Systems: Internals and Design Principles, 5/E*. Prentice Hall, 2005. ISBN: 0-13-147954-7.
- [Tan01] Andrew S. Tanenbaum. *Modern Operating Systems, 4/E*. Prentice Hall, 2001. ISBN: 0-13-031358-0.
- [Tan06] Andrew S. Tanenbaum. *Operating Systems Design and Implementation, 3/E*. Prentice Hall, 2006. ISBN: 0-13-142938-8.