

Universidad Católica San Pablo (UCSP)
Escuela Profesional de
Ciencia de la Computación
SILABO



CS2S1. Sistemas Operativos (Obligatorio)

1. Información general

1.1 Escuela	:	Ciencia de la Computación
1.2 Curso	:	CS2S1. Sistemas Operativos
1.3 Semestre	:	6 ^{to} Semestre.
1.4 Prerrequisitos	:	CS221. Arquitectura de Computadores. (3 ^{er} Sem)
1.5 Condición	:	Obligatorio
1.6 Modalidad de aprendizaje	:	Presencial
1.7 horas	:	2 HT; 4 HP;
1.8 Créditos	:	4
1.9 Plan	:	Plan Curricular 2016

2. Profesores

Titular

- Julio Omar Santisteban Pablo <jsantisteban@ucsp.edu.pe>
 - Doctor en Ciencias de la Computación, Universidad Nacional de San Agustín, Perú, 2021.
 - Master en Internetworking, University of Technology, Australia, 2008.

3. Fundamentación del curso

Un Sistema Operativo es un programa que actúa como intermediario entre el usuario y la máquina. El propósito de un sistema operativo es proveer un ambiente en que el usuario pueda ejecutar sus aplicaciones. En este curso se estudiará el diseño del núcleo de los sistemas operativos. Además el curso contempla actividades prácticas en donde se resolverán problemas de concurrencia y se modificará el funcionamiento de un pseudo Sistema Operativo.

4. Resumen

1. Visión general de Sistemas Operativos 2. Principios de Sistemas Operativos 3. Concurrencia 4. Planificación y despacho 5. Manejo de memoria 6. Seguridad y protección 7. Máquinas virtuales 8. Manejo de dispositivos 9. Sistema de archivos 10. Sistemas *embedded* y de tiempo real 11. Tolerancia a fallas 12. Evaluación del desempeño de sistemas

5. Objetivos Generales

- Conocer los elementos básicos del diseño de los sistemas operativos.

6. Contribución a los resultados (*Outcomes*)

Esta disciplina contribuye al logro de los siguientes resultados de la carrera:

- 1) S.O. Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (**Evaluar**)
- 4) S.O. Reconocer las responsabilidades profesionales y hacer juicios informados en el campo profesional de computación con principios éticos. (**Familiarizarse**)
- 5) S.O. Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (**Usar**)
- 6) S.O. Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. . (**Usar**)
- 7) S.O. Desarrollar tecnología computacional buscando el bien común, aportando con formación humana, capacidades científicas, tecnológicas y profesionales para solucionar problemas sociales de nuestro entorno. (**Evaluar**)

7. Contenido

UNIDAD 1: Visión general de Sistemas Operativos (3)

Competencias: 1

Contenido

- Papel y el propósito del sistema operativo.
- Funcionalidad de un sistema operativo típico.
- Los mecanismos de apoyo modelos cliente-servidor.
- Cuestiones de diseño (eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad)
- Influencias de seguridad, creación de redes, multimedia, sistemas de ventanas.

Objetivos Generales

- Explicar los objetivos y funciones de un sistema operativo moderno [Familiarizarse]
- Analizar las ventajas y desventajas inherentes en el diseño de un sistema operativo [Evaluar]
- Describir las funciones de un sistema operativo contemporáneo respecto a conveniencia, eficiencia, y su habilidad para evolucionar [Familiarizarse]
- Discutir acerca de sistemas operativos cliente-servidor, en red, distribuidos y cómo se diferencian de los sistemas operativos de un solo usuario [Familiarizarse]
- Identificar amenazas potenciales a sistemas operativos y las características del diseño de seguridad para protegerse de ellos [Familiarizarse]

Lecturas: Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)

UNIDAD 2: Principios de Sistemas Operativos (6)	
Competencias: 1	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Estructuración de Sistemas Operativos (monolítico, capas, modular, los modelos micro-kernel) • Abstracciones, procesos y recursos. • Los conceptos de interfaces de programa de aplicación (API) • La evolución de las técnicas de hardware / software y las necesidades de aplicación • Organización de dispositivos. • Interrupciones: métodos e implementaciones. • Concepto de estado de usuario / sistema y la protección, la transición al modo kernel. 	<ul style="list-style-type: none"> • Explicar el concepto de una capa lógica [Familiarizarse] • Explicar los beneficios de construir capas abstractas en forma jerárquica [Familiarizarse] • Describir el valor de la API y <i>middleware</i> [Familiarizarse] • Describir como los recursos computacionales son usados por aplicaciones de software y administradas por el software del sistema [Familiarizarse] • Contrastar el modo <i>kernel</i> y modo usuario en un sistema operativo [Evaluar] • Discutir las ventajas y desventajas del uso de procesamiento con interrupciones [Familiarizarse] • Explicar el uso de una lista de dispositivos y el controlador de colas de entrada y salida [Familiarizarse]
Lecturas: Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 3: Concurrency (9)	
Competencias: 1	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Diagramas de estado. • Estructuras (lista preparada, bloques de control de procesos, y así sucesivamente) • <i>Dispatching</i> y cambio de contexto. • El papel de las interrupciones. • Gestionar el acceso a los objetos del sistema operativo de forma atómica. • La implementación de primitivas de sincronización. • Problemas de multiprocesador (spin-locks, reentrada) 	<ul style="list-style-type: none"> • Describir la necesidad de concurrencia en el marco de un sistema operativo [Familiarizarse] • Demostrar los potenciales problemas de tiempo de ejecución derivados de la operación simultánea de muchas tareas diferentes [Usar] • 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 [Familiarizarse] • Explicar los diferentes estados por los que una tarea debe pasar y las estructuras de datos necesarias para el manejo de varias tareas [Familiarizarse] • Resumir las técnicas para lograr sincronización en un sistema operativo (por ejemplo, describir como implementar semáforos usando primitivas del sistema operativo.) [Familiarizarse] • Describir las razones para usar interrupciones, <i>dispatching</i>, y cambio de contexto para soportar concurrencia en un sistema operativo [Familiarizarse] • Crear diagramas de estado y transición para los problemas de dominios simples [Usar]
Lecturas: Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 4: Planificación y despacho (6)	
Competencias: 1	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • <i>Scheduling preemptive y non-preemptive.</i> • <i>Scheduling</i> y políticas. • Procesos y subprocesos. • Plazos y cuestiones en tiempo real. 	<ul style="list-style-type: none"> • 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 [Evaluar] • Describir las relaciones entre los algoritmos de <i>scheduling</i> y dominios de aplicación [Familiarizarse] • Discutir los tipos de <i>scheduling</i> en procesadores en de corto, mediano, largo plazo y I/O [Familiarizarse] • Describir las diferencias entre procesos y <i>threads</i> [Familiarizarse] • Comparar y contrastar enfoques estáticos y dinámicos para <i>scheduling</i> en tiempo real [Evaluar] • Discutir sobre la necesidad de <i>preemption</i> y <i>deadline scheduling</i> [Familiarizarse] • 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 [Familiarizarse]
Lecturas: Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 5: Manejo de memoria (6)	
Competencias: 1	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Revisión de la memoria física y hardware de gestión de memoria. • Conjuntos de trabajo y thrashing. • El almacenamiento en caché 	<ul style="list-style-type: none"> • Explicar la jerarquía de la memoria y <i>tradeoffs</i> de costo-rendimiento [Familiarizarse] • Resumir los principios de memoria virtual tal como se aplica para el almacenamiento en cache y paginación [Familiarizarse] • Evaluar las ventajas y desventajas en términos del tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y la velocidad del procesador [Evaluar] • Describir las diferentes formas de asignar memoria a las tareas, citando las ventajas relativas de cada uno [Familiarizarse] • 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) [Familiarizarse] • 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 [Familiarizarse]
Lecturas: Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 6: Seguridad y protección (6)	
Competencias:	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Visión general de la seguridad del sistema . • Política / mecanismo de separación. • Métodos de seguridad y dispositivos. • Protección, control de acceso y autenticación. • Las copias de seguridad. 	<ul style="list-style-type: none"> • Explicar la necesidad para la protección y seguridad en un sistema operativo [Familiarizarse] • Resumir las características y limitaciones de un sistema operativo usado para proporcionar protección y seguridad [Familiarizarse] • Explicar el mecanismo disponible en un OS para controlar los accesos a los recursos [Familiarizarse] • 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 [Familiarizarse]
Lecturas: Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 7: Máquinas virtuales (6)	
Competencias: 1	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Tipos de virtualización (incluyendo Hardware / Software, OS, Servidor, Servicio, Red) • Paginación y la memoria virtual. • Sistemas de archivos virtuales. • Los Hypervisores. • Virtualización portátil; emulación vs aislamiento. • Costo de la virtualización. 	<ul style="list-style-type: none"> • Explicar el concepto de memoria virtual y la forma cómo se realiza en hardware y software [Familiarizarse] • Diferenciar emulación y el aislamiento [Familiarizarse] • Evaluar virtualización de compensaciones [Evaluar] • Discutir sobre hipervisores y la necesidad para ellos en conjunto con diferentes tipos de hipervisores [Familiarizarse]
Lecturas: Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 8: Manejo de dispositivos (6)	
Competencias: 1	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Características de los dispositivos serie y paralelo. • Haciendo de abstracción de dispositivos. • Estrategias de buffering. • Acceso directo a memoria. • La recuperación de fallos. 	<ul style="list-style-type: none"> • Explique la diferencia clave entre dispositivos seriales y paralelos e identificar las condiciones en las cuales cada uno es apropiado [Familiarizarse] • Identificar la relación entre el hardware físico y los dispositivos virtuales mantenidos por el sistema operativo [Familiarizarse] • Explicar <i>buffering</i> y describir las estrategias para su aplicación [Familiarizarse] • 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 [Familiarizarse] • Describir las ventajas y desventajas de acceso directo a memoria y discutir las circunstancias en cuales se justifica su uso [Familiarizarse] • Identificar los requerimientos para recuperación de errores [Familiarizarse] • Implementar un controlador de dispositivo simple para una gama de posibles equipos [Usar]
Lecturas: Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 9: Sistema de archivos (6)	
Competencias: 1	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Archivos: los datos, metadatos, operaciones, organización, amortiguadores, secuenciales, no secuencial. • Directorios: contenido y estructura. • Los sistemas de archivos: partición, montar / desmontar sistemas de archivos virtuales. • Técnicas estándar de implementación . • Archivos asignados en memoria. • Sistemas de archivos de propósito especial. • Naming, búsqueda, acceso, copias de seguridad. • La bitacora y los sistemas de archivos estructurados (log) 	<ul style="list-style-type: none"> • Describir las decisiones que deben tomarse en el diseño de sistemas de archivos [Familiarizarse] • Comparar y contrastar los diferentes enfoques para la organización de archivos, el reconocimiento de las fortalezas y debilidades de cada uno. [Evaluar] • Resumir cómo el desarrollo de hardware ha dado lugar a cambios en las prioridades para el diseño y la gestión de sistemas de archivos [Familiarizarse] • Resumir el uso de diarios y como los sistemas de archivos de registro estructurado mejora la tolerancia a fallos [Familiarizarse]
Lecturas: Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 10: Sistemas <i>embedded</i> y de tiempo real (6)	
Competencias: 1	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Proceso y programación de tareas. • Los requisitos de gestión de memoria / disco en un entorno en tiempo real. • Los fracasos, los riesgos y la recuperación. • Preocupaciones especiales en sistemas de tiempo real. 	<ul style="list-style-type: none"> • Describir que hace a un sistema un sistema en tiempo real [Familiarizarse] • Explicar la presencia y describir las características de latencia en sistemas de tiempo real [Familiarizarse] • Resumir los problemas especiales que los sistemas en tiempo real presentan, incluyendo el riesgo, y cómo se tratan estos problemas [Familiarizarse]
Lecturas: Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 11: Tolerancia a fallas (3)	
Competencias: 1	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Conceptos fundamentales: sistemas fiables y disponibles. • Redundancia espacial y temporal. • Los métodos utilizados para implementar la tolerancia a fallos. • 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. 	<ul style="list-style-type: none"> • Explicar la importancia de los términos tolerancia a fallos, fiabilidad y disponibilidad [Familiarizarse] • Explicar en términos generales la gama de métodos para implementar la tolerancia a fallos en un sistema operativo [Familiarizarse] • Explicar cómo un sistema operativo puede continuar funcionando después de que ocurra una falla [Familiarizarse]
Lecturas: Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

UNIDAD 12: Evaluación del desempeño de sistemas (3)	
Competencias: 1	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • ¿Por qué el rendimiento del sistema debe ser evaluado? • ¿Qué se va a evaluar? • Sistemas de políticas de rendimiento, por ejemplo, el almacenamiento en caché, de paginación, la programación, la gestión de memoria, y la seguridad. • Modelos de evaluación: analítica, simulación, o de implementación específico determinista. • Cómo recoger los datos de evaluación (perfiles y mecanismos de localización) 	<ul style="list-style-type: none"> • Describir las medidas de rendimiento utilizados para determinar cómo el sistema funciona [Familiarizarse] • Explicar los principales modelos de evaluación utilizados para evaluar un sistema [Familiarizarse]
Lecturas: Avi Silberschatz (2012), Stallings (2005), Tanenbaum (2006), Tanenbaum (2001), Anderson and Dahlin (2014)	

8. Metodología

1. El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.
2. El profesor del curso presentará demostraciones para fundamentar clases teóricas.
3. El profesor y los alumnos realizarán prácticas
4. Los alumnos deberán asistir a clase habiendo leído lo que el profesor va a presentar. De esta manera se facilitará la comprensión y los estudiantes estarán en mejores condiciones de hacer consultas en clase.

9. Evaluar

Evaluación Continua 1 : 20 %

Examen parcial : 30 %

Evaluación Continua 2 : 20 %

Examen final : 30 %

References

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