UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

FACULTAD DE CIENCIAS MATEMÁTICAS

SÍLABO

1. INFORMACIÓN GENERAL

1.1 Curso	:	Sistemas Operativos
1.2 Código	:	CS2S1
1.3 Programa Académico	:	Computación Científica
1.4 Créditos	:	5
1.5 Prerrequisitos	:	CS210. Algoritmos y Estructuras de Datos. (4^{to} Sem)
1.6 Tipo de curso	:	Electivo
1.7 Área de Estudios	:	Estudios Generales
1.8 Número de Semanas	:	16
1.9 Horas por semana	:	3 HT; 4 HP;
1.10 Ciclo	:	I
1.11 Modalidad de aprendizaje	:	Presencial
1.12 Periodo Académico	:	2026-I
1.13 Inicio del semestre	:	
1.14 Fin del semestre	:	

2. PROFESORES

Atención previa coordinación con el profesor

3. INTRODUCCIÓN AL CURSO

El sistema operativo es una colección de servicios necesarios para interconectar de manera segura el hardware con las aplicaciones. Este curso explora los mecanismos y políticas necesarios para virtualizar el cómputo, la memoria y las E/S. Los estudiantes comprenderán conceptos centrales como concurrencia, persistencia y límites de confianza, los cuales son fundamentales para desarrollar sistemas de software robustos y eficientes.

4. OBJETIVOS

- Comprender el rol y propósito de los sistemas operativos modernos.
- Dominar la planificación de procesos y el ciclo de vida de las unidades de ejecución.
- Analizar la gestión de memoria, incluyendo la virtualización de memoria física y virtual.
- Evaluar sistemas de almacenamiento y técnicas de gestión de archivos.
- Comprender los conceptos básicos de seguridad y protección dentro de un entorno de sistema operativo.

5. RESULTADOS DEL ESTUDIANTE

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usage)

ABET-6) Habilidad para desempeñarse efectivamente en equipos que establecen metas, planifican tareas, cumplen con los plazos y analizan el riesgo y la incertidumbre. (Usage)

6. TEMAS

Unidad 1: Rol y Propósito de los Sistemas Operativos (8 horas)	
Resultados esperados: ABET-2,ABET-6	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Los sistemas operativos median entre el hardware de propósito general y el software específico de la aplicación. ▪ Funciones universales del sistema operativo (por ejemplo, proceso, interfaces de usuario y dispositivos, persistencia de datos) ▪ Funciones extendidas y/o especializadas del sistema operativo (por ejemplo, sistemas embebidos, tipos de servidor como archivos, web, multimedia, gestores de arranque y seguridad de arranque) ▪ Problemas de diseño (por ejemplo, eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad, energía, seguridad, compensaciones entre verificación de errores y rendimiento, flexibilidad y rendimiento, y seguridad y rendimiento). Ver también: Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad ▪ Influencias de seguridad, redes, multimedia, computación paralela y distribuida ▪ Preocupación general de seguridad/protección: Omitir considerar la seguridad en cada capa crea una oportunidad para acceder de manera inapropiada a los recursos. • Se puede lograr acceso no autorizado a archivos en una unidad no cifrada moviendo el medio a otra computadora. • La seguridad impuesta por los sistemas operativos puede ser derrotada infiltrando la capa de arranque antes de que se cargue el sistema operativo. • El aislamiento de procesos puede ser subvertido por una verificación de autorización inadecuada en los límites de la API. • Las vulnerabilidades en el firmware del sistema pueden proporcionar vectores de ataque que eviten por completo el sistema operativo. • El aislamiento inadecuado de la memoria, cómputo y hardware de la máquina virtual puede exponer el sistema anfitrión a ataques desde sistemas invitados. • El sistema operativo puede necesitar mitigar la explotación de vulnerabilidades de hardware y firmware, lo que lleva a posibles reducciones de rendimiento (por ejemplo, mitigaciones de Spectre y Meltdown). ▪ Exposición de funciones del sistema operativo en shells y programación de sistemas. Ver también: Scripting de Shell	▪ Comprender los objetivos y funciones de los sistemas operativos modernos [Explicar] ▪ Evaluar los problemas de diseño en diferentes escenarios de uso (por ejemplo, sistema operativo en tiempo real, móvil, servidor) [Evaluar] ▪ Comprender las funciones de un sistema operativo contemporáneo con respecto a conveniencia, eficiencia y capacidad de evolucionar [Explicar] ▪ Comprender cómo la evolución y la estabilidad son deseables y mutuamente antagónicas en las funciones del sistema operativo [Explicar]
Lecturas : [**Silberschatz2018**], [**Tanenbaum2014**]	

Unidad 2: Scripting de Shell (8 horas)	
Resultados esperados: ABET-2,ABET-6	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Manejo de errores/excepciones ▪ Tuberías (piping) ▪ Comandos del sistema: • Interfaz con sistemas operativos ▪ Variables de entorno ▪ Abstracción de archivo y operadores ▪ Estructuras de datos, como arreglos y listas. ▪ Expresiones regulares ▪ Programas y procesos ▪ Flujo de trabajo	▪ Crear y ejecutar scripts automatizados para gestionar varias tareas del sistema [Crear] ▪ Resolver varios problemas de procesamiento de texto mediante scripting [Resolver]
Lecturas : [**Robbins2005**], [**Silberschatz2018**]	

Unidad 3: Planificación (16 horas)	
Resultados esperados: ABET-2,ABET-6	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Planificación con expropiación y sin expropiación ▪ Planificadores y políticas (por ejemplo, primero en llegar primero en ser servido, trabajo más corto primero, prioridad, round robin, multinivel). Ver también: Gestión de Recursos ▪ Conceptos de planificación de multiprocesador simétrico (SMP) y coherencia de caché Ver también: Jerarquía de Memoria ▪ Temporizadores (por ejemplo, construir muchos temporizadores a partir de temporizadores de hardware finitos). Ver también: Organización de Máquina a Nivel de Ensamblador ▪ Justicia e inanición ▪ Subtemas de sistemas operativos como planificación consciente de energía y planificación en tiempo real. Ver también: Rendimiento y Eficiencia Energética, Características y Arquitecturas Embebidas, Tiempo Real, Memoria y Verificación, Sistemas, Programación y Sensores, Plataformas Móviles: Capacidades y Computación Asíncrona, Plataformas Móviles: Arquitectura y Restricciones, Plataformas Móviles: Fundamentos del Desarrollo ▪ Planificación cooperativa, como futexes de Linux y planificación en espacio de usuario.	▪ Comparar y contrastar los algoritmos comunes utilizados para planificación con y sin expropiación de tareas en sistemas operativos, como prioridad, comparación de rendimiento y esquemas de participación justa [Comparar] ▪ Explicar las relaciones entre algoritmos de planificación y dominios de aplicación [Explicar] ▪ Explicar las distinciones entre tipos de planificadores de procesador como corto plazo, mediano plazo, largo plazo y E/S [Explicar] ▪ Evaluar un problema o solución para determinar la idoneidad para multiprocesamiento asimétrico y/o simétrico [Evaluar] ▪ Evaluar un problema o solución para determinar la idoneidad como proceso vs hilos [Evaluar] ▪ Listar algunos contextos que se benefician de la planificación con expropiación y con plazo límite [Listar/Enumerar] ▪ Explicar las formas en que la lógica incorporada en los algoritmos de planificación es aplicable a otros mecanismos de sistemas operativos, como primero en llegar primero en ser servido o prioridad para E/S de disco, planificación de red, planificación de proyectos y problemas más allá de la computación [Explicar]
Lecturas : [**Silberschatz2018**], [**Stallings2017**]	

Unidad 4: Gestión de Memoria (16 horas)	
Resultados esperados: ABET-2,ABET-6	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Revisión de memoria física, traducción de direcciones y hardware de gestión de memoria. Ver también: Jerarquía de Memoria, Matemáticas Discretas ▪ Impacto de la jerarquía de memoria incluyendo concepto de caché, búsqueda en caché y almacenamiento en caché por CPU en mecanismos y políticas del sistema operativo Ver también: Jerarquía de Memoria, Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento ▪ Direccionamiento lógico y físico, virtualización de espacio de direcciones. Ver también: Jerarquía de Memoria, Matemáticas Discretas ▪ Conceptos de paginación, reemplazo de páginas, trashing y asignación de páginas y marcos ▪ Técnicas de asignación/desasignación/almacenamiento (algoritmos y estructura de datos) rendimiento y flexibilidad • Arenas, asignadores de losas (slab allocators), listas libres, clases de tamaño, páginas de tamaño heterogéneo (páginas enormes) ▪ Almacenamiento en caché de memoria y coherencia de caché y el efecto de vaciar la caché para evitar vulnerabilidades de ejecución especulativaVer también: Organización Funcional, Jerarquía de Memoria, Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento) ▪ Mecanismos y conceptos de seguridad en gestión de memoria incluyendo sandboxing, protección, aislamiento y vectores relevantes de ataque. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad ▪ Memoria virtual: aprovechando el hardware de memoria virtual para servicios del SO y eficiencia	▪ Explicar jerarquía de memoria y compensaciones costo-rendimiento [Explicar] ▪ Resumir los principios de memoria virtual aplicados a almacenamiento en caché y paginación [Resumir] ▪ Evaluar las compensaciones en términos de tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y velocidad del procesador [Evaluar] ▪ Describir la razón y el uso de la memoria caché (rendimiento y proximidad, cómo las cachés complican el aislamiento y la abstracción de máquina virtual) [Describir] ▪ Codificar/Desarrollar programas eficientes que consideren los efectos del reemplazo de páginas y asignación de marcos en el rendimiento de un proceso y el sistema en el que se ejecuta [Crear] ▪ Explicar cómo se utiliza el hardware para una virtualización eficiente [Explicar]
Lecturas : [**Silberschatz2018**], [**ArpaciDusseau2018**]	

Unidad 5: API de Sistemas de Archivos e Implementación (12 horas)	
Resultados esperados: ABET-2,ABET-6	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Concepto de archivo incluyendo datos, metadatos, operaciones y modo de acceso ▪ Montaje de sistema de archivos ▪ Control de acceso a archivos ▪ Compartición de archivos ▪ Métodos básicos de asignación de archivos, incluida la tabla de asignación enlazada ▪ Estructuras del sistema de archivos que comprenden asignación de archivos incluyendo varias estructuras de directorio y métodos para identificar archivos de manera única (por ejemplo, nombre, identificador o ubicación de almacenamiento de metadatos) ▪ Técnicas de asignación/desasignación/almacenamiento (algoritmos y estructura de datos) impacto en rendimiento y flexibilidad (es decir, fragmentación interna y externa y compactación) ▪ Gestión de espacio libre como usar tablas de bits vs enlazado ▪ Implementación de directorios para segmentar y rastrear ubicación de archivos	▪ Explicar las elecciones que deben tomarse en el diseño de sistemas de archivos [Explicar] ▪ Evaluar diferentes enfoques para organización de archivos, reconociendo las fortalezas y debilidades de cada uno [Evaluar] ▪ Aplicar constructores de software apropiadamente dado el conocimiento de la implementación del sistema de archivos [Aplicar]
Lecturas : [**Silberschatz2018**], [**Stallings2017**]	

Unidad 6: Protección y Seguridad (12 horas)	
Resultados esperados: ABET-2,ABET-6	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Descripción general de mecanismos de seguridad del sistema operativo. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad ▪ Ataques y antagonismo (planificación, etc.). Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad ▪ Revisión de vulnerabilidades principales en sistemas operativos reales. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad ▪ Estrategias de mitigación del sistema operativo como copias de seguridad. Ver también: Confiabilidad del Sistema ▪ Separación política/mecanismo. Ver también: Gestión de Riesgos y Datos, Gobierno, Gestión y Política de Seguridad ▪ Métodos y dispositivos de seguridad. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad • Anillos de protección (historia desde Multics hasta x86 virtualizado) • Anillos -1 y -2 de x86_64 (hipervisor y ME/PSP) ▪ Protección, control de acceso y autenticación. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad, Protocolos, Aplicaciones y Criptografía Avanzada, Algoritmos Criptográficos, Fundamentos de Criptografía	▪ Comprender el requisito de mecanismos de protección y seguridad en sistemas operativos [Explicar] ▪ Listar y describir los vectores de ataque que aprovechan vulnerabilidades del SO [Listar/Enumerar] ▪ Comprender los mecanismos disponibles en un SO para controlar el acceso a recursos [Explicar] ▪ Resumir las características y limitaciones de un sistema operativo que impactan la protección y seguridad [Resumir]
Lecturas : [**Silberschatz2018**], [**Tanenbaum2014**]	

7. PLAN DE TRABAJO

7.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

7.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

7.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

8. SISTEMA DE EVALUACIÓN

********* EVALUATION MISSING ********

9. BIBLIOGRAFÍA BÁSICA