

NOMBRE DEL CURSO: Sistemas Operativos 1

CÓDIGO:	281	CRÉDITOS:	5
ESCUELA:	Ciencias y Sistemas	ÁREA A LA QUE PERTENECE:	Ciencias de la Computación.
PRE REQUISITO:	781 – Org. Lenguajes Compiladores 2. 778 – Arq. Compu y Ensambladores 1	POST REQUISITO:	285 Sistemas Operativos 2 775 Sistemas de Bases de Datos 2
CATEGORÍA:	Obligatorio	SEMESTRE:	2do. Semestre 2022
CATEDRÁTICO (A):	Ing. Sergio Arnaldo Méndez Aguilar	AUXILIARES:	German José Paz Cordon
EDIFICIO:	Virtual	SECCIÓN:	N
SALÓN DEL CURSO:	Salón virtual asignado en Portal de Facultad de Ingeniería	SALÓN DEL LABORATORIO:	Salón virtual asignado en el Portal de Facultad de Ingeniería
HORAS POR SEMANA DEL CURSO:	4 períodos	HORAS POR SEMANA DEL LABORATORIO:	2 períodos
DÍAS QUE SE IMPARTE EL CURSO:	Lunes y Miércoles	DÍAS QUE SE IMPARTE EL LABORATORIO:	Miércoles
HORARIO DEL CURSO:	18:10 – 19:20	HORARIO DEL LABORATORIO:	15:40 – 17:20

DESCRIPCIÓN DEL CURSO:

El curso de sistemas operativos busca que el estudiante comprenda de forma práctica y sencilla los aspectos a evaluar en los sistemas operativos, teniendo como punto de vista el estudio de este a través de su funcionamiento en la administración de procesos y la máquina extendida o máquina virtual. Adicionalmente busca introducir al estudiante a las prácticas de administración de sistema, manejo de paralelismo y concurrencia.

OBJETIVO GENERAL:

Proporcionar al estudiante los conocimientos necesarios para entender el comportamiento y construcción de un sistema operativo desde el punto de vista de estudio de un sistema operativo como administrador de procesos y como máquina extendida.

OBJETIVOS ESPECÍFICOS:

1. Entender e implementar virtualización y cloud computing, y la concurrencia de procesos.
2. Aprender a administrar procesos en un sistema operativo.

3. Implementar sistemas que implementan concurrencia y paralelismo, como optimización de servidores usando tecnologías asociadas a dichos conceptos.
4. Implementar Brokers para administrar eventos concurrentes en un sistema.
5. Exponer al estudiante en las últimas tecnologías de servidores con respecto a sistemas con concurrencia y paralelismo, utilizando orquestadores de contenedores y tecnologías cloud native.

METODOLOGÍA:

El desarrollo del curso se basa en el funcionamiento y administración de un sistema operativo estudiando sus componentes a través del estudio del sistema operativo como administrador de procesos y como máquina extendida.

Distribución de Puntos

Actividad	Ponderación
Proyecto Fase 1	10 %
Proyecto Fase 2	30 %
Proyecto Fase 3	30 %
2 Prácticas	10 %
Tareas, investigaciones, exámenes cortos, exposiciones	10 %
Participación eventos Cloud Native	5 %
Examen Final	5 %
Total	100 %

PROGRAMATICO Y CALENDARIZACIÓN:

CONTENIDO:

Unidad 1: Máquina extendida

- 1.1 Creación de scripts en Bash
- 1.2 Virtualización por hardware con KVM
- 1.3 Cloud Computing con OpenStack
- 1.4 Virtualización de sistema operativo: Containers
- 1.5 Containers con Docker y Containerd
 - 1.5.1 Creación de containers en x86_64, ARMv7 y ARM64
 - 1.5.2 Técnicas de cross compiling en Rust y Go
 - 1.5.3 Dockerfiles multi stage
 - 1.5.4 Imágenes distroless
 - 1.5.5 Reducción de tamaño de imágenes con Alpine
- 1.6 Google Cloud como cloud provider
 - 1.6.1 Virtual Machines con architectures x86_64 y ARM
 - 1.6.2 Cloud Run
 - 1.6.3 Cloud Functions
 - 1.6.4 Container registry
 - 1.6.5 Load Balancers

Unidad 2: Procesos y programación concurrente

- 2.1 Manejo de procesos en Linux
 - 2.1.1 Comandos para administración de Procesos
 - 2.1.2 Uso de /proc para capturar información de procesos
- 2.2 Diagrama de transición de procesos
- 2.3 Módulos de Kernel
 - 2.3.1 Comandos para administración de módulos de Kernel
 - 2.3.2 Creación de módulos en Linux
 - 2.3.3 Dispositivos virtuales en /dev
 - 2.3.4 System Calls

2.4 Exclusión mutua e Interbloqueo

2.5 PCB

2.6 Introducción a Kubernetes

Unidad 3: Concurrencia, paralelismo y sistemas distribuidos

- 3.1 Uso de Hilos en Rust
- 3.2 Manejo de concurrencia usando Go, channels y goroutines
- 3.3 Operaciones síncronas y asíncronas
 - 3.3.1 Operaciones síncronas y asíncronas en Rust con Tokio
- 3.4 Comunicación de procesos usando gRPC
- 3.5 Concurrencia en bases de datos NoSQL
 - 3.5.1 Uso de MongoDB para almacenar documentos
 - 3.5.2 Redis para Caches y manejo de información
- 3.6 Implementación Brokers con Kafka y RabbitMQ

BIBLIOGRAFIA

- Sistemas Operativos (William Stallings)
- Contenido Sistemas Operativos de Universidad Rutgers, Paul Krzyzanowski, New Jersey