| | |
|---|---|
| Nombre de la materia: | **LABORATORIO DE SISTEMAS OPERATIVOS** |
| Clave: | **CI7200-L** |
| No. De horas /semana : | **5** |
| Duración semanas: | **16** |
| Total de Horas : | **80** |
| No. De créditos : | **10** |
| Prerrequisitos : | **Ninguno** |

## Objetivo:

El estudiante conocerá la estructura y funcionalidad externa e interna de un sistema operativo, los recursos que administra el sistema operativo, las técnicas que se utilizan para la administración de estos recursos, la interacción entre los componentes del sistema operativo y los aspectos mas importantes que influyen en el desempeño de los diferentes componentes del sistema operativo.

## Metodología del curso:

El alumno implementará simulaciones de sistema operativo y con esto se enfrentará a los retos de Ingeniería en el área de desarrollo de un sistema operativo. Esta experiencia servirá para obtener un conocimiento profundo en el área y así poder definir estrategias de solución viables o resolución eficiente a los diversos problemas que pueden presentarse en el ramo de desarrollo y administración de sistemas.

Se recomienda que el lenguaje de programación para los proyectos sea C pues es el lenguaje por opción para programar sistemas operativos.

## Contenido:

| | | |
|---|---|---|
| **1** | **Simulación básica de un CPU** | **5 horas** |
| | 1.1 Revisión de proyecto | 1 hora |
| **2** | **Administración de procesos.** | **15 horas** |
| | 2.1 Revisión de proyecto | 1 hora |
| **3** | **Planificación del CPU.** | **15 horas** |
| | 3.1 Revisión de proyecto | 1 hora |
| **4** | **Sincronización de procesos.** | **15 horas** |
| | 4.1 Evitación y detección de interbloqueos. | 15 horas |
| | 4.1.1 Revisión de proyecto | 1 hora |
| **5** | **Administración de memoria.** | **30 horas** |
| | 5.1 Asignación contigua por particiones dinámicas. | 15 horas |
| | 5.1.1 Revisión de proyecto | 1 hora |
| | 5.2 Asignación no contigua por paginación. | 15 horas |
| | 5.2.1 Revisión de proyecto | 1 hora |

## Programa desarrollado

1 **Simulación básica de un CPU:** Desarrollar una simulación de instrucciones ensamblador. Un procesador contará con 4 registros de usuario AX, BX, CX y DX, un registro de contador de programa y un registro de instrucciones. Este simulador será capaz de interpretar las siguientes instrucciones: MOV, ADD, SUB, MUL, DIV, INC y DEC, además de una instrucción especial END que indicará el fin del programa. Las instrucciones que el simulador ejecutará serán cargadas desde un archivo de texto proporcionado por el usuario en tiempo de ejecución.

El simulador acepta los comandos:

- ejecutar *archivo:* ejecuta las instrucciones almacenadas en el archivo *archivo*. Mostrando en todo momento al usuario los valores de los registros del CPU simulado.
- salir: sale del simulador.

2 **Administración de procesos:** Basándose en el simulador anterior agregar la capacidad de ejecutar múltiples archivos de instrucciones. El simulador asignará un número máximo de instrucciones que podrá ejecutar cada proceso antes de ser interrumpido y puesto en estado listo, eventualmente el proceso será seleccionado de nuevo y deberá continuar su ejecución de manera transparente. Cabe hacer hincapie en que el simulador permitirá la entrada a sólo un proceso a ejecución y esta ejecución deberá efectuarse sobre los registros de simulación de CPU, cuando un proceso alcance su tiempo máximo de ejecución su contexto será guardado en su PCB y otro proceso será seleccionado.

La selección del siguiente proceso a ejecutar es libre, aunque se recomienda que sea al estilo round-robin por su simpleza.

El simulador en todo momento debe mostrar el estado del CPU y un monitor de procesos indicando: pid, estado, contexto y nombre de la imagen.

Agregar al simulador los siguientes comandos:

- mata *pid*: que mueve al estado salida el proceso cuyo pid sea *pid.* Posteriormente en el estado de salida el simulador realizará las operaciones necesarias de limpieza de proceso.

3 **Planificación del CPU:** Implementar la planificación de CPU en el simulador, el algoritmo a implementar será diferente a Round-Robin y se deja a criterio del profesor.

4 **Sincronización de procesos**

4.1 Evitación y detección de interbloqueos: Se deja a criterio del profesor la decisión de implementar el algoritmo del banquero o el

algoritmo de detección de interbloqueos. Para lograr la petición de recursos, se implementarán las siguientes instrucciones:

- MAX W-X-Y-Z: que incializará los máximos recursos a usar por el proceso. W unidades del primer recurso, X del segundo, etc. Esta debe ser la primera instrucción del proceso.
- GET W-X-Y-Z: que solicita se asignen W unidades del primer recurso, X del segundo, etc. En caso que el simulador otorgue la petición el proceso continuará su ejecución, en caso contrario pasará a un estado suspendido y será activado cuando se pueda satisfacer su petición.
- USA W-X-Y-Z: que hará uso de W unidades del primer recurso, X del segundo, etc. El proceso será bloqueado por un tiempo aleatorio mientras simula E/S con los recursos asignados.
- FRE W-X-Y-Z: libera W unidades del primer recurso, X del segundo, etc.

5 **Administración de memoria.** Para ambos proyectos, se requiere que el simulador tenga un bloque contiguo de almacenamiento en una variable llamada RAM, este bloque fungirá como memoria principal del sistema y será asignada de acuerdo al proyecto. La unidad de almacenamiento es 1 instrucción. Ambos subproyectos utilizan traducción de direcciones.

5.1 Asignación contigua por particiones dinámicas: Se hará uso de la memoria por medio de particiones dinámicas, utilizando técnicas de compactación, relocalización y unión de huecos libres. Además de elegir una estrategia de asignación por primer ajuste, mejor ajuste o peor ajuste. Este proyecto se desarrolla sobre el proyecto 4.

5.2 Asignación no contigua por paginación: La asignación será por medio de páginas de tamaño de 4 instrucciones. El no asignará más de 3 marcos de página a un proceso, esto con el fin de que exista la ejecución de procesos parcialmente cargados así como la implementación de políticas de asignación, acceso, sustitución y ubicación. Este proyecto se desarrolla sobre el proyecto 4 y será el último módulo del proyecto final.

## Bibliografía básica:


William Stallings. Sistemas Operativos: aspectos internos y principios de diseño., 5ta edición Pearson Educación 2005
Milan Milenkovic. Sistemas Operativos: conceptos y diseño. 2da edición. McGraw-Hill 1996
Abraham Silberschatz. Fundamentos de Sistemas Operativos. 7ma edición McGraw-Hill 2006


## Bibliografía complementaria


Andrew S. Tanenbaum. Sistemas operativos modernos, 2da edición. Pearson Educación 2003


## Metodología de enseñanza-aprendizaje:

| | |
|---|---|
| Revisión de conceptos, análisis y solución de problemas en clase: | ( X ) |
| Lectura de material fuera de clase: | ( X ) |
| Ejercicios fuera de clase (tareas): | ( ) |
| Investigación documental: | ( X ) |
| Elaboración de reportes técnicos o proyectos: | ( X ) |
| Prácticas de laboratorio en una materia asociada: | ( ) |
| Visitas a la industria: | ( ) |

## Metodología de evaluación:

| | |
|---|---|
| Asistencia: | ( X ) |
| Tareas: | ( ) |
| Elaboración de reportes técnicos o proyectos: | ( X ) |
| Exámenes de Academia o Departamentales | ( ) |

**Programa propuesto por M.C. Luis Eduardo Gamboa Guzmán**