




Supuesto Didáctico

Módulo: Programación – 1.º DAM

Simulación de técnicas de entrada/salida (E/S)




2.1 ¿Qué supuesto queremos trabajar?

Simular mediante programación las tres técnicas clásicas de E/S:

-  E/S programada (polling)
-  E/S con interrupciones
-  DMA (acceso directo a memoria)

 Y comparar su impacto en el rendimiento del sistema.

Aprendizajes esperados

Dimensión	Contenido
 Conceptos	Técnicas de E/S, uso de CPU, eficiencia del sistema
 Procedimientos	Diseño de simulaciones, uso de temporizadores, medición de tiempos
 Actitudes	Razonamiento lógico, curiosidad técnica, autonomía y análisis comparativo

2.2 Contextualización del alumnado

- Curso: 1.º DAM
- Módulo: Programación
- Momento: mitad de curso, con conocimientos previos en estructuras de control
- Nivel técnico: intermedio en lógica algorítmica y funciones
- Interés: acercar programación a fenómenos reales del sistema operativo

2.3 Conocimientos previos requeridos

- Bucles `while` , `for`
- Temporización (`sleep()` o similar)
- Uso de funciones definidas por el usuario
- Lógica condicional y estructura secuencial
- Conceptos básicos sobre uso de CPU y multitarea

2.4 Objetivos de aprendizaje

- Simular y comparar las técnicas de E/S
- Medir y analizar el impacto en el uso de CPU
- Visualizar el comportamiento mediante gráficos
- Desarrollar pensamiento algorítmico aplicado a sistemas
- Fomentar autonomía técnica en pequeños proyectos

2.5 Metodología

- ◆ Aprendizaje basado en la experimentación
- ◆ Desarrollo guiado de simulaciones
- ◆ Comparación de resultados y reflexión grupal
- ◆ Incorporación de elementos visuales para reforzar conceptos
- ◆ Trabajo individual o en parejas, con apoyo docente

2.6 Material didáctico (DUA)

- Código base con estructura inicial
- Esquemas explicativos de cada técnica
- Plantillas de simulación por bloques
- Tablas comparativas vacías para rellenar
- Recursos extra para profundizar (documentación, vídeos, herramientas)

2.7 Secuencia de acciones formativas

1. Introducción teórica: ¿Qué es la E/S?
2. Análisis de las tres técnicas (polling, interrupciones, DMA)
3. Diseño del simulador con `sleep()` , `contador` , `tiempo`
4. Medición del uso de CPU en cada técnica
5. Representación gráfica (barras, línea temporal)
6. Discusión y comparativa de resultados
7. Presentación de conclusiones y entrega

2.8 Actividad principal

Simulación comparativa de técnicas de E/S

El alumnado debe desarrollar un programa que:

- Simula **una operación de entrada/salida**
- Mide el uso o bloqueo de la CPU en tres escenarios:
 - ◆ **E/S programada** → la CPU espera activamente
 - ◆ **E/S con interrupciones** → la CPU hace tareas mientras espera
 - ◆ **DMA** → el dispositivo transfiere los datos sin intervención de la CPU

 Representación visual:

- Gráficos de uso de CPU
- Líneas temporales o barras para comparar eficiencia



2.9 Evaluación: Instrumentos y criterios




Criterio	Instrumento
Precisión en la simulación	Pruebas funcionales
Claridad de visualización	Gráficos o salida organizada
Uso de estructuras correctas	Revisión de código
Análisis de resultados	Reflexión escrita/oral
Creatividad y ampliaciones	Valoración opcional

2.10 Inclusión y atención a la diversidad


- Código base con ejemplos paso a paso
- Flexibilidad en el nivel de detalle o representación
- Posibilidad de trabajar en modo texto o visual
- Tutoría técnica individual para seguimiento
- Recursos explicativos en múltiples formatos (visual, escrito)

2.11 Actividades de ampliación

Para alumnado avanzado:

-  Simulación de planificación de E/S en disco (FCFS vs SCAN)
-  Introducir aleatoriedad en peticiones de E/S
-  Comparar resultados con herramientas reales:
 - `iostat` , `htop` , `dstat`

Cierre

- ✓ Relación directa con los contenidos de sistemas operativos
 - ✓ Ejercicio integrador de programación y eficiencia de recursos
 - ✓ Fomenta pensamiento técnico y análisis de comportamiento real
-  ¡Haz que tu simulador piense como un sistema operativo!