

Sistema de Memoria Virtual para Multiprocesamiento

Diseño del Sistema De La Tarea
Sistemas Operativos

Integrantes:

Camila Rodríguez - C36624

Yosery Zheng Lu - C38680

24 de noviembre, 2025

1. Introducción

Este documento presenta el diseño de un sistema de memoria virtual que simula multiprocesamiento mediante `fork()`. El sistema traduce direcciones lógicas a físicas usando una MMU, maneja page faults automáticamente, y emplea el algoritmo Enhanced Second Chance para reemplazo de páginas. Características principales: 4 procesos concurrentes, 512 bytes de RAM (32 frames \times 16 bytes), TLB de 4 entradas, y soporte para programas con loops y jumps.

2. Especificaciones Técnicas

- Memoria Física: 512 bytes (32 frames \times 16 bytes)
- Memoria Virtual por Proceso: 1 KB (64 páginas \times 16 bytes)
- Procesos Concurrentes: 4 máximo (creados con `fork`)
- Direccionamiento Lógico: 10 bits (VPN: 6 bits, Offset: 4 bits)
- Direccionamiento Físico: 9 bits (Frame: 5 bits, Offset: 4 bits)

Page Table Entry (PTE):

- V (Valid): 1 bit - Página en memoria
- D (Dirty): 1 bit - Página modificada
- R (Reference): 1 bit - Acceso reciente
- Frame: 5 bits - Número de frame (0-31)

TLB: 4 entradas con algoritmo LRU

Algoritmo de Reemplazo: Enhanced Second Chance (Clock)

Estrategia: Global (pool común de frames)

3. Arquitectura y Componentes

El sistema consta de:

- VirtualMemorySystem: Coordina procesos, MMU y memoria.
- Process: Cada proceso (fork) tiene su PCB con registros R0-R9, PC, flags, y su propia PageTable de 64 entradas.
- CPU: Ejecuta instrucciones fetch-decode-execute. Cada acceso a memoria pasa por la MMU.
- MMU: Traduce direcciones lógicas a físicas. Flujo: (1) Consulta TLB, (2) Si TLB miss, consulta PageTable, (3) Si V=0, maneja page fault.
- PageTable: 64 entradas por proceso. Cada PTE tiene bits V, D, R y frame number.
- PhysicalMemory: 32 frames de 16 bytes. Gestiona frames libres y coordina reemplazo.
- TLB: Caché de 4 entradas (VPN→Frame) con LRU. Acelera traducciones 10-100x.
- EnhancedSecondChance: Clasifica páginas en 4 clases según bits R y D:
 - Clase 0 (R=0, D=0): Mejor víctima
 - Clase 1 (R=0, D=1): Segunda opción
 - Clase 2 (R=1, D=0): Dar segunda oportunidad
 - Clase 3 (R=1, D=1): Peor víctima
- Disk: Almacena programas en archivos .txt. Carga páginas en page faults.
- Sincronización: Mutexes POSIX para MMU, RAM, y pool de frames.

Ver diagramas: diagrama_arquitectura.png, diagrama_clases_simple.png, diagrama_traduccion.png, diagrama_algoritmo_clock.png

4. Algoritmo de Reemplazo

- Enhanced Second Chance recorre frames circularmente (clock hand) buscando:
 - Primera pasada: Buscar Clase 0 ($R=0$, $D=0$). Si se encuentra, seleccionar.
 - Segunda pasada: Buscar Clase 1 ($R=0$, $D=1$). Limpiar bit R de Clase 2 y 3.
 - Tercera pasada: Buscar Clase 0 (algunas fueron degradadas).
- Si víctima tiene $D=1$, escribir a disco antes de reemplazar.
- Ventajas vs FIFO: Considera uso reciente y modificación.
- Ventajas vs LRU: Más simple, sin timestamps, overhead mínimo.

5. Justificación de Decisiones

- Tamaño de memoria y páginas:
 - 512 bytes: Suficiente para demostrar conceptos, fuerza page faults
 - 16 bytes/página: Balance entre fragmentación y overhead. 4 instrucciones por página.
- Enhanced Second Chance:
 - FIFO no permitido. Enhanced considera R y D para mejor rendimiento.
 - Aproxima LRU con menor overhead.
- Estrategia Global:
 - Mejor utilización total de memoria.
 - Procesos con más demanda obtienen más frames dinámicamente.
- 4 Procesos:
 - Excede mínimo de 2.
 - $32 \text{ frames} / 4 = 8 \text{ frames promedio por proceso}$ (suficiente).
 - Demuestra competencia real por recursos.
- TLB (Puntos Extra):
 - 4 entradas capturan localidad temporal.

- LRU simple con 4 entradas.
- Tasa de hit esperada: 80-90% en loops.
- Reporta estadísticas hits/misses.

6. Clases Principales

- VirtualMemorySystem: processes[], mmu*, memory*
- Process: pid, page_table*, cpu*, program
- CPU: registers[10], pc, flags
- MMU: tlb*, page_tables map
- PageTable: entries[64], process_id
- PageTableEntry: valid, dirty, reference, frame
- PhysicalMemory: memory[512], frames[32], replacement*
- TLB: entries[4], lru_counter[]
- EnhancedSecondChance: clock_hand