

Relatório - Trabalho 3

1. Introdução

O objetivo do Trabalho 3 foi implementar o subsistema de gerenciamento de memória virtual. Isso incluiu a simulação de uma MMU (*Memory Management Unit*), tabelas de páginas, tratamento de falta de página (*Page Fault*), memória secundária (Disco) e algoritmo de substituição de páginas (LRU). O sistema agora suporta processos cujo tamanho excede a memória física disponível e protege a memória entre processos distintos.

2. Carga de Programas e Espaço de Endereçamento

A função `so_carrega_programa` copia o código do executável para a **Memória Secundária (Disco)**. O processo é criado com a tabela de páginas vazia (Paginação sob Demanda). Foi necessário distinguir entre o endereço físico e o endereço virtual. Correções foram feitas para garantir que o `regPC` fosse inicializado com o endereço virtual, enquanto o SO mantinha o rastreamento da posição no disco.

3. Tratamento de Page Fault (*trata_page_fault*)

O fluxo lógico desenvolvido foi:

1. Identificar o endereço virtual que causou a falha (lendo o registrador complemento salvo pela CPU).
2. Verificar se existe um quadro livre na RAM (*busca_quadro_livre*).
3. **Substituição de Página (Swap-out)**: Se a memória estiver cheia, o algoritmo LRU escolhe uma página "vítima". Se ela foi modificada, é escrita de volta no disco. A tabela de páginas do dono da vítima é atualizada (marcada como inválida).
4. **Carga da Página (Swap-in)**: A página necessária é lida do disco para o quadro liberado.
5. **Bloqueio por Latência**: Para simular o hardware real, o processo é colocado em estado BLOQUEADO por um tempo definido (`TEMPO_DISCO`), simulando a lentidão do disco mecânico.

4. Algoritmo LRU com Envelhecimento

Implementou-se o algoritmo de envelhecimento para aproximar o LRU.

- A cada interrupção de relógio, um contador associado a cada página é deslocado para a direita (divisão por 2).
- Se a página foi acessada (bit de acesso = 1), soma-se um valor alto no bit mais significativo.
- A página com o menor valor no contador é a que foi usada "menos recentemente" e torna-se a candidata à substituição.

5. Conclusão

O Trabalho 3 adicionou uma camada de complexidade significativa ao simulador. A implementação da memória virtual, proteção de memória e outros conceitos extremamente relevantes foi crucial para sua compreensão. A introdução da latência de disco (`TEMPO_DISCO`) evidenciou a importância de um escalonador eficiente para manter a CPU ocupada enquanto processos aguardam operações de memória lentas. Também foi perceptível um incremento grande no tempo total de execução dos processos, justamente devido às buscas em disco.