MAC0422 - Sistemas Operacionais

Daniel Macêdo Batista

IME - USP, 30 de Novembro de 2020

Roteiro

Acelerando a paginação

Acelerando a paginação

Acelerando a paginação

- Duas ações são críticas com paginação:
  - Mapeamento do endereço virtual para o endereço físico (tem que ser rápido)
  - Tamanho da tabela de páginas (Se o espaço de endereços virtuais é grande, vai ser grande a tabela de páginas)
- O mapeamento tem que ser rápido porque é feito a cada referência à memória (várias por instrução)
- O tamanho de páginas é crítico porque com o avanço dos computadores, o tamanho do espaço de endereços virtuais tende a crescer (mais páginas)

Primeira tentativa

- Ter a tabela de páginas como um conjunto de registradores muito rápidos
- Cada registrador = 1 página virtual, indexada pelo número de página virtual
- Quando um processo inicia → o SO carrega esses registradores com a tabela de páginas do processo (copia da memória principal).
- **Problema**: não resolve o problema do tamanho da tabela de página. Se é muito grande, a solução ficaria muito cara financeiramente (registrador é uma memória cara)
- **Problema**: não resolve por inteiro o problema de velocidade porque a cada mudança de contexto terá que atualizar os registradores

Segunda tentativa – TLB

- Esquema muito usado por vários SOs
- Relembrando que com a paginação, a tabela de página estará na memória. Assim, cada acesso à memória na verdade será pelo menos 2. Um para acessar a tabela de páginas e outro para de fato acessar a posição real na memória física
- Relembrando do *working set*, programas tendem a referenciar muitas vezes um número pequeno de páginas (apenas algumas entradas da tabela de páginas são muito referenciadas)
- A solução então vai ser permitir o mapeamento de uma pequena quantidade de páginas (preferencialmente o *working set*) sem necessidade de ir na tabela de páginas. Assim, evita dois acessos à memória cada vez que for necessário referenciar uma posição na memória virtual

Segunda tentativa – TLB

- Precisa de um hardware para isso, esse hardware é a TLB (*Translation Lookaside Buffer*).
- Geralmente fica dentro da própria MMU
- Tem poucas entradas (8-64)
- Cada entrada: bit de validade da página, número da página virtual, bit de modificado, bits de proteção (RWX) e o quadro da página física
- Resolve o problema de velocidade (não vai fazer dois acessos)
- Resolve o problema de espaço (não vai colocar a tabela de páginas inteira aqui)
- Não fica tão caro financeiramente (são apenas poucos bytes. Não é a tabela de páginas inteira)

Segunda tentativa – TLB

- Quando um endereço virtual chegar na MMU para tradução, verifica primeiro se está na TLB comparando todas simultaneamente (importante ter um hardware que permita a comparação simultânea)
- Se encontra na TLB e se o tipo de acesso está permitido pelos bits de permissão, lê o endereço do quadro de página e faz o acesso (sem necessidade de ir na tabela de páginas)

Segunda tentativa – TLB

- Se não encontra na TLB, funciona como já vimos no funcionamento da MMU
- Escolhe alguma entrada para remover da TLB e põe no lugar o mapeamento recém realizado nessa entrada

Terceira tentativa – TLB

- Fazer o que a TLB faz mas via software
- O SO vai manter a TLB, ao invés de um hardware especial para isso na MMU
- A vantagem é que não ocupa espaço no hardware que pode ser usado pela CPU
- Estando no nível do SO ele pode usar a "intuição" para tentar pré-carregar algumas páginas