Arquitetura de Computadores 2018/19

TPC 5

Prazo de entrega: 23:59 de 11 de junho de 2019

Tópicos: *Endereços virtuais e páginas.*

Este trabalho de casa consiste em dois exercícios a resolver individualmente. Pode discutir ideias gerais com colegas, mas a solução e a escrita do código deve ser estritamente individual. A entrega será via Mooshak do DI. Todas as resoluções serão comparadas de forma automática e os casos de fraude serão punidos de acordo com os regulamentos em vigor.

NOTA: só são aceites as primeiras 10 submissões ao mooshak de cada problema, sendo avaliada a com mais pontos no mooshak. Se submeter mais vezes, serão ignoradas!

Introdução

São fornecidos dois programas que simulam a transformação de endereços numa MMU para uma arquitetura com memória paginada (rever aulas 23 e 24 e o cap. 9 do livro recomendado). O primeiro programa, `mmu1.c`, calcula a transformação de endereços virtuais em endereços reais com base numa tabela de páginas. O segundo programa, `mmu2.c`, é semelhante ao primeiro, mas simula uma MMU mais completa que faz uso de uma TLB (*translation lookaside buffer*).

Os ficheiros contendo a tabela de páginas e os endereços virtuais a transformar, são ficheiros de texto com os endereços representados em hexadecimal. Em particular, os ficheiros dos endereços são iguais aos usados no TPC anterior. Os nomes destes ficheiros devem ser indicados na linha de comandos quando executa o simulador, para que este os leia. O simulador inicia a tabela de páginas e, depois, efetua a respetiva simulação das transformações dos endereços nos acessos à memória real.

A arquitetura simulada possui endereços virtuais e reais de 32 bits e páginas/frames de 8 Kbytes.

A simulação é feita nas seguintes condições:

- Só há um processo, cuja tabela de páginas e acessos a memória são os descritos nos ficheiros indicados na linha de comandos.
- Existe sempre memória real para todas as páginas do processo (a tabela de páginas tem todas as entradas necessárias).

Exercício 1 (50%)

Neste exercício deve bastar completar a implementação da função `translateOneAddr` em `mmu1.c`. Esta converte um endereço virtual no real (físico). Para tal deve consultar a tabela de páginas (`pageTable`) para obter a respetiva página física (*frame*) e calcular o respetivo endereço real.

Para testar a sua solução, execute comandos como o seguinte:

```
mmu1 bziptab bzip.trace
```

Deve obter nas primeiras linhas (e última) os seguintes valores:

```
Page table size: 512000
23a0
4d40
4d60
23c0
7360
8308
...
Memory accesses: 1000000
```

Para o seguinte exemplo, mais simples, "`mmu1 pagtab test.trace`", deve obter:

```
Page table size: 3
20000
20002
20100
61000
b000
Memory accesses: 5
```

Neste exercício submeta ao mooshak apenas o seu ficheiro **`mmu1.c`**.

Exercício 2 (50%)

Este programa simula as ações do *hardware* de paginação, com TLB, incluindo as atualizações da TLB. A simulação é feita nas seguintes condições:

- A MMU tem uma TLB do tamanho indicado na função `main`, e que fica na variável `TLBsize`.
- Quando se consulta a tabela de páginas, coloca-se essa entrada na TLB, substituindo outra mais antiga se necessário. A substituição será por uma ordem FIFO, ou seja, sobrepõe a entrada mais antiga.

Sugestão: pode obter esta ordem usando um índice que tenha sempre a próxima posição da TLB a usar para novas entradas. Assim, este começa em zero e vai avançando pela TLB. Quando chegar ao fim da TLB, volta ao início substituindo as entradas mais antigas (existe um índice `nextToTLB` no código fornecido que pode ser usado para esse fim).

Note que as caches reais, tal como a TLB, costumam usar políticas semelhantes a LRU e não FIFO, como aqui.

Deve contar o número de TLB hits (variável `hitTLB`) para que no fim o programa afixe a taxa de hits. Por exemplo, para executar o simulador para o ficheiro `bzip.trace`, deve escrever o seguinte comando (note que deve obter exatamente os mesmos endereços reais que na versão anterior):

```
      mmu2 bziptab bzip.trace
```

```
Page table size: 512000
23a0
4d40
4d60
23c0
7360
...
173180
Memory accesses: 1000000
TLB hits: 998196 (99.8196%)
```

Para o exemplo mais simples, "`mmu2 pagtab test.trace`", deve obter:

```
Page table size: 3
20000
20002
20100
61000
b000
Memory accesses: 5
TLB hits: 2 (40%)
```

Neste exercício submeta ao mooshak apenas o seu ficheiro **`mmu2.c`**