

Arquitetura de Computadores 2017-18

Ficha 11

Tópicos: Caches.

Introdução

A matéria teórica ilustrada nesta aula prática corresponde às aulas teóricas sobre caches, em particular as de mapa directo. A matéria encontra-se também nos capítulos 6 e 9 do livro *Computer Systems: A Programmer's Perspective 2nd Ed.*

Todo o código necessário para este trabalho é fornecido via CLIP. Para compilar, basta dar o comando `make` no terminal. Este usa as definições em `Makefile` para (re)compilar apenas os ficheiros necessários e produzir um novo executável devidamente atualizado.

Simulação de cache de mapa direto

Obtenha a partir do CLIP o ficheiro `cachesim.zip` que contém um simulador do *hardware* que acede à cache para cada endereço de memória acedido. O simulador recebe dois argumentos:

1. O número de linhas na cache a usar na simulação; e
2. Um ficheiro com um traço (registo) dos endereços de memória acedidos durante a execução de um programa.

No simulador considera-se que as linhas da cache guardam blocos de memória de 32 bytes. O ficheiro com o traço dos endereços de memória acedidos é um ficheiro de texto com o seguinte formato (exemplo):

```
0041f7a0 R
31348900 W
```

Em cada linha, o primeiro valor é um inteiro (hexadecimal) com o endereço de memória a que o CPU acede. A letra representa o tipo de operação realizada naquele endereço de memória: R (leitura) ou W (escrita). Estão disponíveis dois ficheiros que contêm sequências de referências à memória feitas por dois programas reais, nomeadamente:

```
gcc.trace.gz      O compilador GNU CC
bzip.trace.gz     O compressor BZIP
```

O programa fornecido simula as ações do *hardware* que permitem decidir se cada acesso terá sucesso na cache (hit) ou não (miss). Para tal, em `cache.c` deve completar a função `simulateOneStep` que deve verificar se determinado endereço está ou não na cache, devolvendo `HIT` ou `MISS`, consoante o caso e, a quando de um *miss*, atualiza a cache de acordo com o funcionamento das caches de mapa direto. Os ficheiros fornecidos são os seguintes:

<code>Makefile</code>	Ficheiro de configuração para o comando <code>make</code>
<code>simulador.c</code>	Contém o programa principal do simulador
<code>cache.h</code> <code>cache.c</code>	Contêm o código que simula o funcionamento da cache (a completar)
<code>stats.h</code> <code>stats.c</code>	Contêm o código para coligir e imprimir as estatísticas do processo de simulação

Uma vez completada a implementação e compilado o programa, pode executar uma simulação indicando o tamanho da cache e o ficheiro com o traço de execução. Exemplo:

```
simulador 64 gcc.trace
```

Para este exemplo, deve obter o seguinte resultado:

```
CACHE SIZE = 2048 (64 lines*32 bytes)
INPUT FILE = gcc.trace
number of memory references = 1000000
number of cache misses = 363274
hitratio = 63.7%
```

Com este simulador analise a evolução do *hitratio* com o tamanho da cache, por exemplo para os casos de 16, 64, 256, 512, 1024 linhas (e mais). Confirme que mesmo esta organização de cache (mapa direto) consegue uma relativamente elevada taxa de hits, mesmo com caches não muito grandes. Observe também que os dois programas têm localidades bastante diferentes e, assim, a mesma cache revela taxas de sucesso diferentes para cada programa.