

Arquitectura de Computadores I

Modo priverligiado, Excepções, Input/Output

Miguel Barão

Níveis privilegiados de execução

Podem existir vários níveis de privilégios. Nos níveis mais altos, existem mais registos e instruções para controlar a máquina.

User (U) nível mais baixo de privilégios, usado para aplicações.

Supervisor (S) usado para o sistema operativo.

Machine (M) nível mais elevado, dá acesso completo à máquina.
Para código de total confiança.

Podem existir vários níveis de privilégios. Nos níveis mais altos, existem mais registos e instruções para controlar a máquina.

User (U) nível mais baixo de privilégios, usado para aplicações.

Supervisor (S) usado para o sistema operativo.

Machine (M) nível mais elevado, dá acesso completo à máquina.
Para código de total confiança.

O modo M é obrigatório em todas as implementações da arquitectura.

- Microcontroladores muito simples só têm o modo M.
- Processadores em geral suportam os modos U, S e M.

Os nossos programas correm no nível U (único simulado no RARS).

A mudança para níveis mais elevados de privilégios dá-se pelo mecanismo de exceções:

Exception é um evento síncrono referente a uma condição anormal que ocorre durante a execução de uma instrução.

Interrupt é um evento assíncrono que pode causar uma transferência de controlo inesperada.

Trap é a transferência de controlo de controlo para uma *trap handler* causada por uma exceção ou interrupção.

- Load word de endereço não múltiplo de 4 → **excepção!**
- Movimento do rato → **interrupção!**
- Rotina do sistema operativo que responde às interrupções/excepções → **trap handler!**

- Load word de endereço não múltiplo de 4 → **exceção!**
- Movimento do rato → **interrupção!**
- Rotina do sistema operativo que responde às interrupções/exceções → **trap handler!**

Quando ocorre uma *trap*:

- 1 a instrução em execução é interrompida,
- 2 a causa e origem da exceção/interrupção é registada em registos especiais,
- 3 o nível de privilégio aumenta
- 4 a execução continua num *trap handler*.

A arquitectura prevê a existência um número elevado de registos, mas nem todos são obrigatoriamente implementados.

Alguns exemplos de registos que são automaticamente escritos quando ocorre uma trap:

Machine Exception Program Counter (mepc) guarda o endereço da instrução que foi interrompida.

Machine Cause Register (mcause) indica o evento que causou a trap.

Machine Status Register (mstatus) indica e controla o modo de operação do processador.

Lista de Exceções

Num.	Nome
0	instruction-address-misaligned
1	fetch-address
2	illegal-instruction
3	breakpoint
4	misaligned-load
5	load-access
6	misaligned-store
7	store-access
8	user-ecall
9	supervisor-ecall
10	hypervisor-ecall
11	machine-ecall
12	pagefault-instruction
13	pagefault-load
15	pagefault-store

A instrução **ecall** é usada para realizar um pedido de um serviço a um nível mais privilegiado.

A exceção gerada depende do modo em que a instrução é executada:

- Modo U - gera a exceção 8 (user-ecall)
- Modo S - gera a exceção 9 (supervisor-ecall)
- Modo M - gera a exceção 11 (machine-ecall)

Por exemplo, um programa a correr no modo U pede um serviço ao sistema operativo (*system call*) com a instrução **ecall**, que delega o tratamento da exceção ao modo S.

- `uret` retorna de uma trap para o modo U
- `sret` retorna de uma trap para o modo S
- `mret` retorna de uma trap para o modo M
- `wfi` gestão de interrupções
- `sfence.vma` gestão de memória em modo S
- `hfence.bvma` gestão de memória em modo H
- `hfence.gvma` gestão de memória em modo H