Excepções

ASC I 2015/2016

Teresa Gonçalves

Resumo

- Excepções, Traps e Interrupções
- Mecanismo de Excepções (User e Kernel Modes)
- System Calls
- Espaço de Endereçamento

Excepções, Traps e Interrupções

Excepções, Traps e Interrupções interrompem o fluxo normal de execução de um programa para correr rotinas (*exception handlers*) do sistema operativo.

Excepção é uma condição que ocorre *excepcionalmente* na execução do código. Exemplos: load de um endereço inválido, overflow numa operação aritmética.

Trap é explicitamente colocada num programa usando instruções especialmente concebidas para esse efeito. Ex.: `teqi $t0, 2 # traps if equal immediate`

Interrupção é tipicamente provocada por um dispositivo externo. Exemplos: movimento do rato, temporizador.

Exemplos de excepções

Arithmetic Overflow ocorre quando a soma de dois números não pode ser representada correctamente nos registos disponíveis.

- A instrução `add` faz uma soma, detecta overflow e provoca uma excepção caso ocorra.
- A instrução `addu` faz uma soma, mas não detecta overflow e não gera nenhuma excepção.

Address Error ocorre nas seguintes situações:

- fetch de um endereço não alinhado (não múltiplo de 4).
- load ou store de um endereço não alinhado.
- referência a um endereço do espaço de endereçamento Kernel, quando o processador está em User Mode.

Exemplos de excepções

Trap ocorre quando uma instrução de *trap* resulta em true e gera a excepção.

System Call ocorre quando é executada a instrução `syscall`.

Reserved Instruction ocorre quando é executada uma instrução (código máquina) que não está definida na arquitectura ou implementada.

Floating Point ocorre quando o coprocessador de vírgula flutuante (CP1) pretende sinalizar uma excepção deste tipo.

Interrupt é tipicamente gerada por hardware.

Exemplo: overflow

```
    # Suponhamos que
    # t0 = 0x7fffffff
    # t1 = 1
    add $t2, $t0, $t1
```

A soma resulta em 0x80000000, que é negativo: Overflow.

É gerada uma excepção, que consiste em:

- Passar o processador para Kernel Mode.
- Modifica vários registos específicos registando a ocorrência (Registos do CP0 - CoProcessador de Controlo).
- Salta para o endereço 0x80000180.

A instrução addu é semelhante a add, mas não gera excepção.

Exemplo: address error

```
# Suponhamos que
# t0 = 0x10000003
lw $t1, 0($t0)
```

O endereço `0x10000003` não é alinhado: Address Error
(endereço não é múltiplo de 4)

É gerada uma excepção, que consiste em:

- Passar o processador para Kernel Mode.
- Modifica vários registos específicos registando a ocorrência (Registos do CP0 - CoProcessador de Controlo).
- Salta para o endereço `0x80000180`.

Chamadas ao sistema: syscall

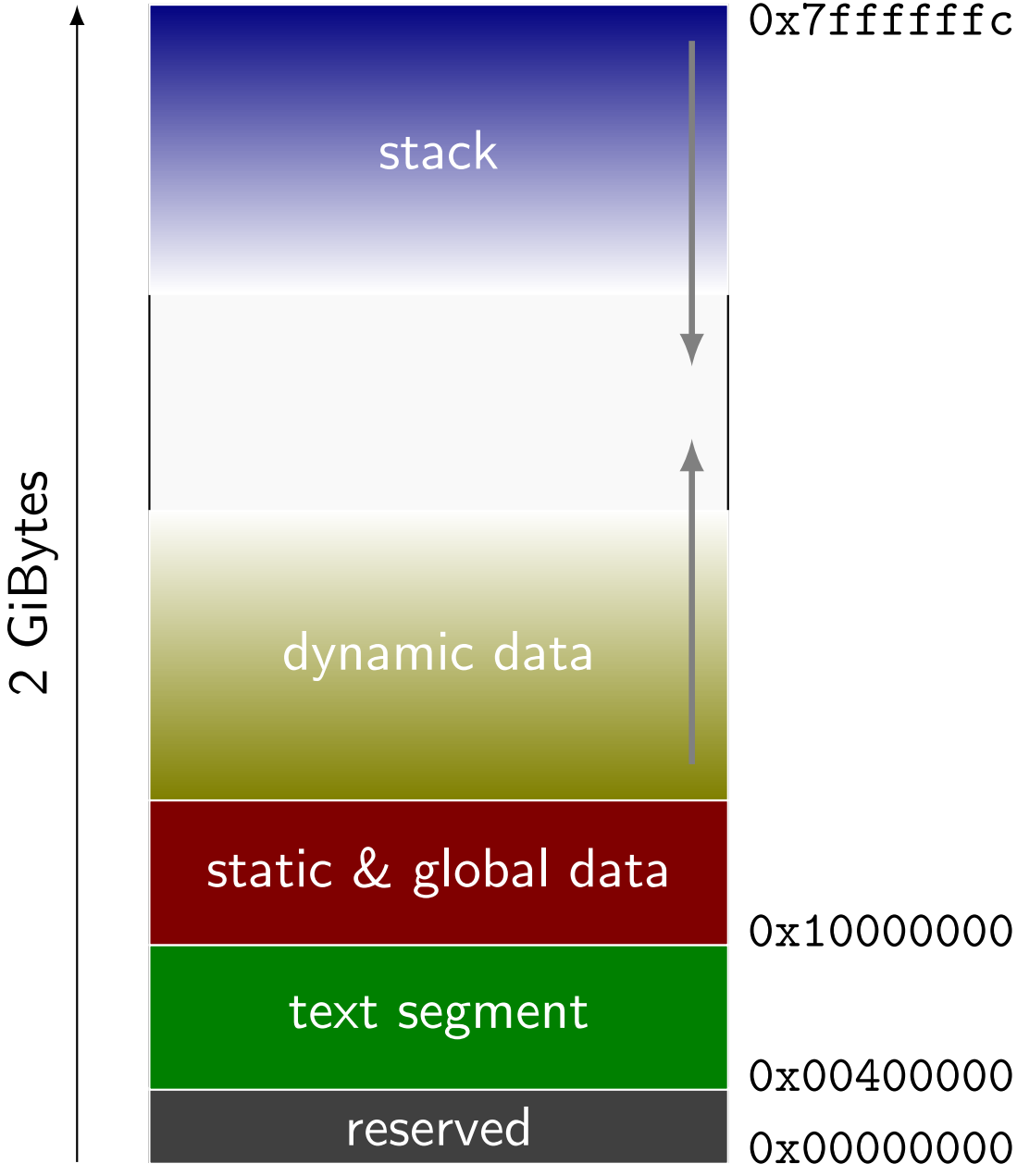
- Um programa em execução em *User Mode* não tem acesso a I/O (*Input/Output*).
- A comunicação com periféricos apenas pode ser efectuada em *Kernel Mode* pelo Sistema Operativo.
- Um programa em *User Mode* pede um serviço ao Sistema Operativo, *System Call*, gerando uma excepção.
- A instrução `syscall` é específica para este fim.

```
li $a0, 15      # argumento
li $v0, 1       # serviço print_int
syscall         # system call (excepcao)
```

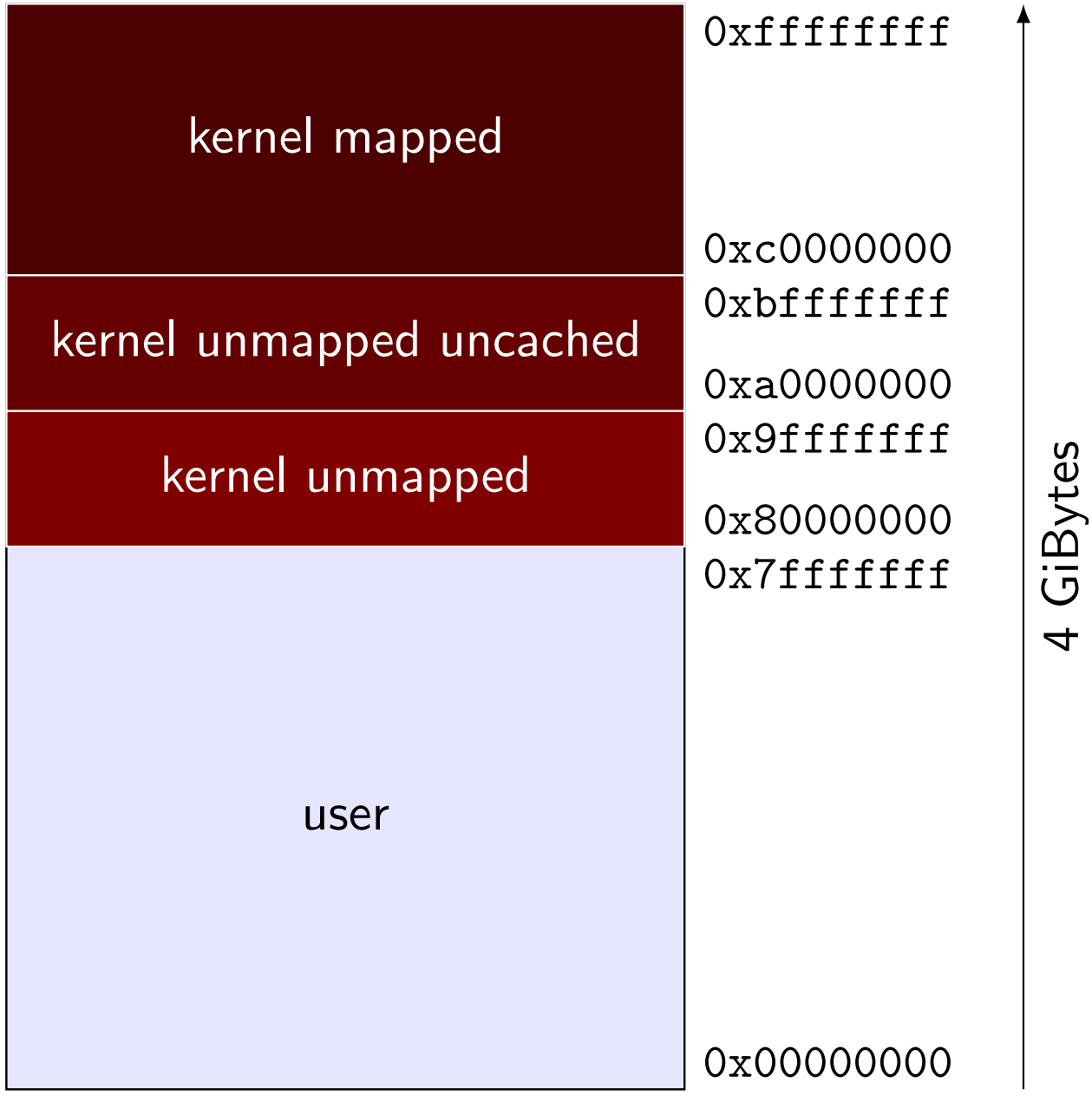
System calls dos simuladores MARS e SPIM

Os simuladores MARS e SPIM suportam um conjunto de *systems calls*.

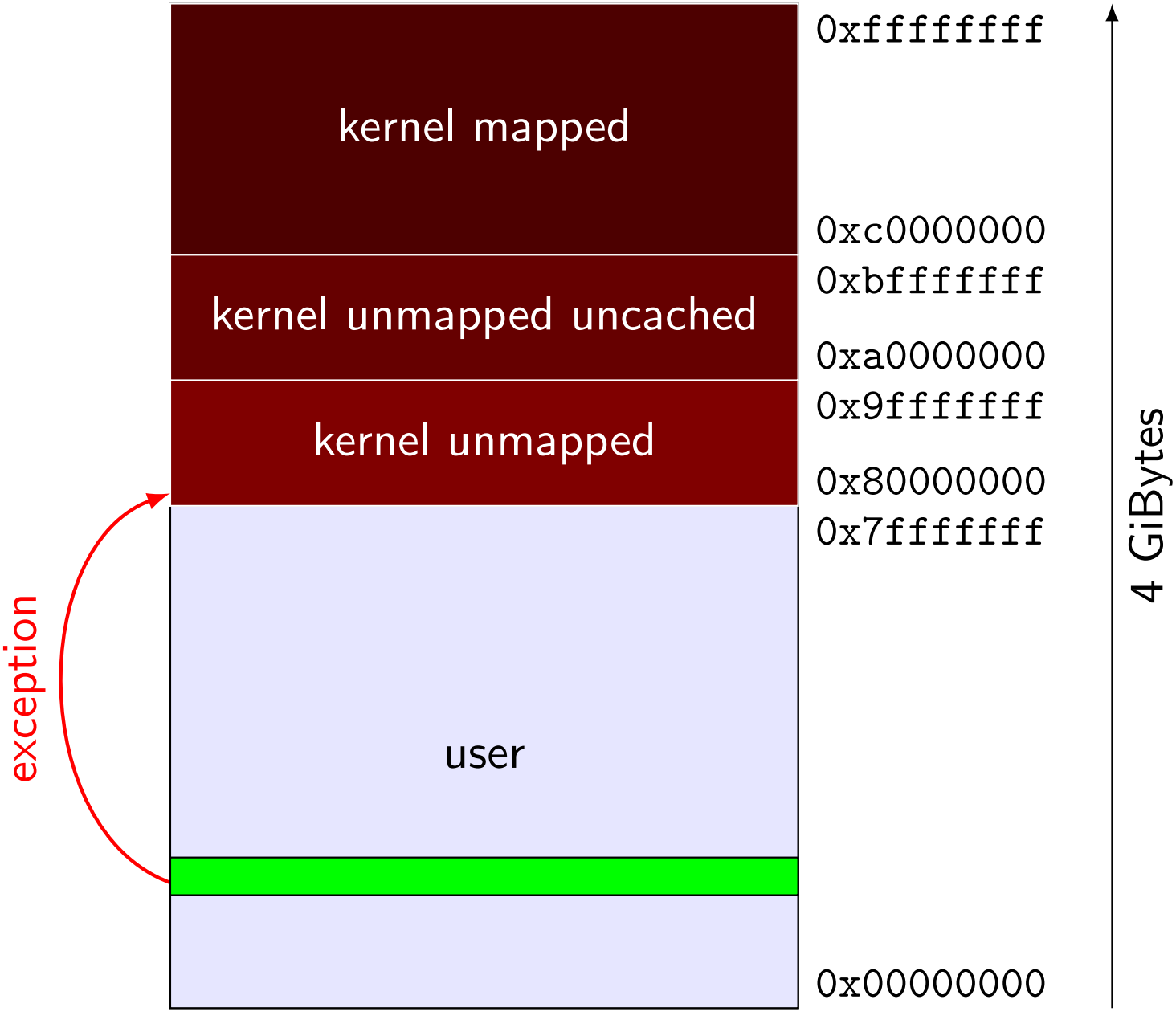
Espaço de endereçamento em User Mode



Espaço de endereçamento completo



Espaço de endereçamento completo



Exemplos

O que faz a seguinte função?

```
f:
    add $a0, $a0, $a1
    li $v0, 1
    syscall

    jr $ra
    nop
```

Que excepções podem ser geradas por esta função?

Exemplos

O que faz a seguinte função?

```
f:
    add $a0, $a0, $a1
    li $v0, 1
    syscall

    jr $ra
    nop
```

Que excepções podem ser geradas por esta função?
[*Resposta: Overflow e Syscall*]

Exemplos

O que faz a seguinte função?

```
g:
    li $v0, 1
L:
    syscall
    addiu $a0, $a0, -1
    bne $a0, $zero, L
    nop
    jr $ra
    nop
```

Que excepções podem ser geradas por esta função?

Kernel Mode

Em kernel mode o processador tem acesso a um conjunto maior de registos, instruções especiais, e acesso a todo o espaço de endereçamento sem restrições. Alguns dos registos do CP0 são:

BadVAddr (CP0 reg. 8) *bad virtual address* é um registo só de leitura que contém o último endereço virtual que causou uma das excepções AddressError, TLBRefill, TLBInvalid ou TLBModified.

Status (CP0 reg. 12) é um registo de leitura/escrita que contém o modo de operação (user/kernel), activação/desactivação de interrupções, e estados de diagnóstico do processador.

Cause (CP0 reg. 13) é um registo só de leitura que contém a causa da última excepção nos bits 6 a 2. Contém mais informação que não se descreve aqui.

EPC (CP0 reg. 14) *exception program counter* é um registo de leitura/escrita que contém o endereço do programa a retomar a execução normal após uma excepção ou serviço do S.O. ter sido processada.

Exemplo de um registo do coprocessador de controlo

Status Register (CP0, registo 12):

31-28	27	26	25	24	23	22	21	20	19	18	17-16	15-10	9-8	7	6	5	4	3	2	1	0
CU3-CU0	RP	FR	RE	MX	PX	BEV	TS	SR	NMI	0	Impl	IPL	IM1-IM0	KX	SX	UX	UM	R0	ERL	EXL	IE

CU3-CU0 controla acesso aos coprocessadores 3,2,1 e 0. Bit 1: acesso permitido, bit 0: acesso negado.

RP activa modo de poupança de energia (*reduced power*).

FR activa suporte de números em vírgula flutuante de 64 bits nos registos coprocessador de vírgula flutuante.

RE *reverse endianness* quando executa em modo User. Permite alterar o endianness usado no acesso à memória de um programa particular.

MX activa suporte MDMX e DSP.

PX activa acesso a operações de 64 bits num processador MIPS64.

BEV controla *exception vectors* (pontos de entrada na ocorrência de excepções normais ou boot do sistema).

TS TLB detectou várias linhas iguais (*matches*) durante uma escrita na TLB.

SR Excepção foi devida a um *soft reset*.

NMI Excepção foi devida a uma NMI (*nonmaskable interrupt*).

IPL *Interrupt priority level*. Uma interrupção é efectivada apenas se tiver um nível superior ao indicado aqui.

IM1-IM0 *Interrupt mask*.

KX,SX,UX Relacionados com acesso a espaço de endereçamento de 64 bits em processadores de 64 bits nos modos Kernel, Supervisor e User.

KSU Modo de operação do processador indica se está num dos três modos: Kernel, Supervisor ou User mode.

ERL *Error level* é activado em certas excepções: Reset, Soft Reset, NMI ou CacheError.

EXL *Exception level* é activado com todas as excepções excepto as anteriores (do ERL).

IE *Interrupt Enable* funciona como interruptor geral para activar ou desactivar o sistema de interrupções.