

Homework 2

钟赞 2016K8009915009

2018 年 10 月 8 日

1. 列出一种指令系统的不同运行级别之间的关系

以MIPS处理器为例，运行级别主要包括三种：调试模式、根模式和客户模式，根模式和客户模式又分别包含核心模式、监管模式和用户模式（但监管模式很少使用）。所有运行模式相互独立，即处理器在某一时刻只能处于某一种运行模式下。处理器上电复位后处于根-核心模式，随后根据需要转换到其他模式下。

不同运行模式可控制的处理器资源不同。根-核心模式和调试模式都可以控制所有的处理器资源，根-用户模式和客户-用户模式只能控制各自模式自身的资源，客户-核心模式可以控制所有客户-用户模式的资源。

2. 用c语言描述段页式存储管理管理的地址转换过程

转换过程如下图：

```
struct address{
    Type_seg SegmentNum;           //段号
    Type_page PageNum;             //页号
    Type_off Offset;               //页内偏移
}*addr_vir;                       //初始的虚拟地址

Type_addr segment_addr;           //段表基地址
Type_width page_width;           //页号的位数
page_addr = *(segment_addr + addr_vir->SegmentNum); //页表基地址
//计算物理地址：物理页号+页内偏移
addr_phy= *(page_addr + addr_vir->PageNum) << page_width + addr_vir->Offset;
```

3. 请简述桌面电脑PPT翻页过程中用户态和核心态的转换过程

我们按一下键盘使PPT翻页，键盘会产生信号送到南桥芯片，南桥芯片把键盘的编码保存在一个寄存器中，并向处理器发送一个外部中断信号。该外部中断信号传到CPU内部后把其中一个称为Cause的控制寄存器的某一位置为“1”，表示收到了外部中断。CPU中另外一个称为Status的控制寄存器有屏蔽位来确定是否处理这个外部中断信号。

屏蔽处理后的中断信号被附在一条译码后的指令上送到重排序缓冲。外部中断是例外（也称“异常”）的一种，发生例外的指令不会被送到功能部件执行。当这条指令成为重排序缓冲的第一条指令时CPU处理例外。

重排序缓冲向所有的模块发出一个取消信号，取消该指令后面的所有指令：修改控制寄存器，把系统态设为核心态；保存例外原因、发生例外的程序计数器等到制定的控制寄存器中；然后把程序计数器的值置为0x80000180这个例外处理入口地址进行取指。

0x80000180是MIPS结构制定的操作系统例外处理入口地址。处理器跳转到0x80000180后执行操作系统代码，操作系统首先保存处理器现场，包括寄存器的内容等。保存现场后，操作系统向CPU的控制寄存器读例外原因，发现是外部中断例外，就向南桥的中断控制器读中断原因，读的同时清除南桥的中断位。读回来后发现中断原因是有人敲了空格键。

操作系统接下来要查找读到的空格是谁的：有没有进程处在阻塞状态等键盘输入。操作系统发现有一个名为PowerPoint的进程处于阻塞态，这个进程对空格键会有所响应，就把PowerPoint唤醒。

PowerPoint被唤醒后处在运行状态。发现操作系统传过来的数据是个键盘输入空格，表示要翻页。PowerPoint就把下一页要显示的内容准备好，调用操作系统中的显示驱动程序，把要显示的内容送到显存，由图形处理器通过访问显存空间刷新屏幕。达到了翻一页的效果。

4. 给定下列程序片段：

$$A = B + C$$

$$B = A + C$$

$$C = B + A$$

- (1) 写出上述程序片段在四种指令系统类型（堆栈型、累加器型、寄存器-存储器型、寄存器-寄存器型）中的指令序列。

堆栈型	累加器型	寄存器-存储器型	寄存器-寄存器型
PUSH C PUSH B ADD POP A POP B PUSH A ADD POP B POP A POP C PUSH B PUSH A ADD POP C	LOAD B ADD C STORE A ADD C STORE B ADD A STORE C	LOAD R1,B ADD R1,C STORE A,R1 ADD R1,C STORE B,R1 ADD R1,A STORE C,R1	LOAD R1,B LOAD R2,C ADD R3,R1,R2 STORE A,R3 ADD R1,R2,R3 STORE B,R1 ADD R2,R1,R3 STORE C,R2

- (2) 假设四种指令系统都属于CISC型，令指令码宽度为x位，寄存器操作数宽度为y位，内存地址操作数宽度为z位，数据宽度为w位。分析指令的总位数和所有内存访问的总位数。

	指令总位数	内存访问总位数
堆栈型	$x+z$ 或 x	$x+z$
累加器型	$x+z$	$x+z$
寄存器-存储器型	$x+y+z$ 或 $x+z$	$x+y+z$ 或 $x+z$
寄存器-寄存器型	$x+y+z$ 或 $x+3y$	$x+y+z$

(3) 微处理器由32位时代进入64位时代，上述四种指令系统类型哪种更好？

寄存器-寄存器型更好。寄存器访问速度快，便于编译器调度优化，并可以充分运用局部性原理，大量操作可以在寄存器中完成。另外，寄存器之间的相关性容易判断，容易实现流水线、多发射和乱序执行等方法。

5. 写出0xDEADBEEF在大尾端和小尾端下的内存中的排列

内存地址由低到高				
大尾端	DE	AD	BE	EF
小尾端	EF	BE	AD	DE

6. 在你的机器上编写c程序来得到不同的数据类型占用的字节数，给出程序和结果

c程序：

```
#include<stdio.h>
int main()
{
    printf("char: %d\nshort: %d\nint: %d\nlong: %d\nlong
long: %d\n", sizeof(char), sizeof(short), sizeof(int),
sizeof(long), sizeof(long long));
    return 0;
}
```

运行结果：

```
stu@stu-VirtualBox:~/Desktop$ ./test
char: 1
short: 2
int: 4
long: 4
long long: 8
```

7. 根据MIPS指令集的编码格式计算条件转移指令和直接转移指令的跳转范围

直接跳转指令j是J型指令，编码格式为：

opcode(6)	address(26)
-----------	-------------

指令的低26位用于跳转目标的编址，由于指令在内存中是4字节对齐的，因此跳转偏移量的低二位无需存储，为00。所以跳转的寻址范围为 $2^{28} = 256\text{MB}$ 。

条件转移指令b为I型指令，编码格式为：

opcode(6)	rs(5)	rt(5)	immediate(16)
-----------	-------	-------	---------------

指令的低16位为有符号相对跳转偏移量，由于指令是4字节对齐的，偏移量的低二位也为00，故跳转范围为 $2^{18} = 256\text{MB}$ (-128MB~128MB)。

8. 不使用LWL和LWR，写出如图2.10的不对齐加载（小尾端）

	3	2	1	0
内存:	0x22	0x33	0x44	
				0x11

①LW R1, 1

REG R1	0x22	0x33	0x44	
--------	------	------	------	--

②SRL R1, R1, 8

REG R1		0x22	0x33	0x44
--------	--	------	------	------

③SLL R2, 4, 24

REG 2	0x11			
-------	------	--	--	--

④add R1, R1, R2

REG 1	0x11	0x22	0x33	0x44
-------	------	------	------	------