MIPS基础

覃晓婉

MIPS 32个32位通用寄存器

寄存器编号	寄存器名	用法
$0	$zero	常数0
$1	$at	保留给寄存器处理大常数
$2~$3	$v0~$v1	存放函数的返回值
$4~$7	$a0~$a3	存放函数调用参数
$8~$15	$t0~$t7	临时寄存器
$16~$23	$s0~$s7	存储寄存器，可存放变量
$24~$25	$t8~$t9	临时寄存器
$26~$27	$k0~$k1	用于保存中断信息
$28	$gp	全局指针
$29	$sp	栈指针
$30	$fp	帧指针
$31	$ra	子程序调用返回地址

MIPS常用汇编指令

- MIPS汇编指令格式：

标号：操作码 操作数1 操作数2 操作数3 #注释

Label: add $s0,$s1,$s2 #$s0=$s1+$s2

MIPS常用汇编指令

指令类型	指令	例子	含义
算术运算指令	加法	add $s0,$s1,$s2	$s0=$s1+$s2
	减法	sub $s0,$s1,$s2	$s0=$s1-$s2
	加立即数	addi $s0,$s1,10	$s0=$s1+10
数据存入与读取	读出字	lw $s0,offset($s1)	$s0=memory[offset+$s1]
	存入字	sw $s0,offset($s1)	memory[offset+$s1]= $s0
	读出半字	lh $s0,offset($s1)	$s0=memory[offset+$s1]
	读出无符号半字	lhu $s0,offset($s1)	$s0=memory[offset+$s1]
	存储字节	sb $s0,offset($s1)	memory[offset+$s1]= $s0
	读取字节	lb $s0,offset($s1)	$s0=memory[offset+$s1]
	读取立即数到高半字	lui $s0,30	$s0=30*2^16
	读取无符号字节	lbu $s0,offset($s1)	$s0=memory[offset+$s1]

MIPS常用汇编指令

指令类型	指令	例子	含义
逻辑运算	与	and $s0,$s1,$s2	$s0=$s1&$s2
	或	or $s0,$s1,$s2	$s0=$s1|$s2
	或非	nor $s0,$s1,$s2	$s0=~($s1|$s2)
	与立即数	andi $s0,$s1,10	$s0=$s1&10
	或立即数	ori $s0,$s1,10	$s0=$s1|10
	逻辑左移	sll $s0,$s1,10	$s0=$s1<<10
	逻辑右移	srl $s0,$s1,10	$s0=$s1>>10

MIPS常用汇编指令

指令类型	指令	例子	含义
条件跳转	相等转移	beq $s0,$s1,lable	if($s0==$s1) goto lable
	不相等转移	bne $s0,$s1,lable	if($s0 ！=$s1) goto lable
	小于设置	slt $s0,$s1,$s2	if($s1<$s2) $s0=1 else $s0=0
	低于设置	sltu $s0,$s1,$s2	if($s1<=$s2) $s0=1 else $s0=0
	小于常数设置	slti $s0,$s1,20	if($s1<20) $s0=1 else $s0=0
	低于常数设置	sltiu $s0,$s1,10	if($s1<=10) $s0=1 else $s0=0
无条件跳转	直接跳转	j lable	goto lable
	间接跳转	jr $ra	goto $ra
	跳转并连接	jal lable	$ra=PC+4 goto lable

MIPS常用伪指令

伪指令	含义
.rdata	只读数据
.align 2	两字节对齐
.globl status	全局变量status
.set	设置代码的属性
.type @function	定义的是函数，这里的话可以看出是数据还是函数
.set noreorder	不让代码优化
.set reorder	优化代码
.extern g_data	对外部变量的引用
main：	标号main
.set push	将当前的属性保存起来，是的设置的属性仅作用于当前的代码
.set pop	回复之前保存的属性
.text	代码段
.data	数据段
.ent	用于标记函数的起点
.end	用于标记函数结束的位置
.set macro A	定义宏A
.endm	结束宏定义
.ascii “HELLO\0”	定义字符串不附带空结束符
.asciz “HELLO”	定义字符串附带空结束符

MIPS CP0寄存器及指令

- 协处理器CP0起到控制CPU的作用。MMU、异常处理、乘除法等功能，都依赖于协处理器CP0来实现。

- 对CP0的指令：
- mfc0 rt, rd　#将CP0中的rd寄存器内容传输到rt通用寄存器；
- mtc0 rt, rd　#将rt通用寄存器中内容传输到CP0中寄存器rd；

- 如：

```
#define CP0_STATUS $12
mfc0  $k0, CP0_STATUS #把CP0_STATUS的内容拷贝到K0寄存器
```

```
.text
.global main
main:
        # 1) task1 call BIOS print letter ‘a’
        lw $a0,letter
        lw $t2,printch
        jal $t2
.data
letter: .word ‘a’
# 3. PMON print char function address
# printch(char ch)
printch: .word 0x80011140
```

Makefile

- Makefile 文件，告诉make命令需要怎么样的去编译和链接程序

- Makefile的规则:

  target ... : prerequisites ...

  command

```
CC = mipsel-linux-gcc

all: image

bootblock: bootblock.s
	${CC} -G 0 -O2 -fno-pic -mno-abicalls -fno-builtin -nostdinc -mips3 -Ttext=0xffffffffa0800000 -N -o bootblock bootblock.s -nostdlib -e main -Wl,-m -Wl,elf32ltsmip -T ld.script

kernel: kernel.c
	${CC} -G 0 -O2 -fno-pic -mno-abicalls -fno-builtin -nostdinc -mips3 -Ttext=0xffffffffa0800200 -N -o kernel kernel.c -nostdlib -Wl,-m -Wl,elf32ltsmip -T ld.script

createimage:
	gcc createimage.c -o createimage

image: bootblock kernel
	 ./createimage --extended bootblock kernel
clean:
	 rm -rf bootblock image kernel *.o
floppy:
	sudo fdisk -l /dev/sdb
	sudo dd if=image of=/dev/sdb conv=notrunc
```

作业

将下列c程序换成MIPS汇编实现，在课程网站提交代码和设计文档

- 选RISC-V的同学明天晚上之前交，其他选MIPS的同学下周一之前交
- 选RISC-V的同学需要用截图来体现gdb单步调试project1
- 在MIPS汇编中可以使用project1框架里printstr()来实现打印

```
int main()
{
        int i = 1;
        int sum = 0;
        do
        {
                sum = sum + i;
                i = i + 1;
        } while (i <= 100);
        printf("sum =  %d\n",sum);
        return 0;
}
```