**体系结构 第二章**

**陈彦帆 2018K8009918002**

1. MIPS：运行级别主要包括三种：调试模式（Debug Mode）、根模式（Root Mode）和客户模式（Guest Mode），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。所有运行模式互相独立。

其中，根-核心模式和调试模式都可以控制所有的处理器资源，包括改变虚实地址映射关系、控制系统环境和进程切换等。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，可以执行非特权指令，使用通用寄存器和浮点寄存器，所有访存落在一个扁平均一虚地址空间上。客户-核心模式可以控制所有客户-用户模式的资源。

2. //注：代码不完整，待修改

addr\_t transform(addr\_t vaddr)

{

    segment\_t seg = SegmentTable[getSegNum(vaddr)];

    if(seg.length<getPageNum(vaddr))

        error();

    page\_t page = PageTable[seg.base+getPageNum(vaddr)];

    return page.pnum<<PAGEBITS | getOffset(vaddr);

}

3. 完成翻页包括一系列需要系统调用的操作，如读硬盘、显示等。进行系统调用时，程序传递系统调用号和相关参数，发起中断。进入核心态，操作系统保存上下文，跳转到系统调用号对应的地址进行中断处理，等待IO时可能切换到其它进程执行，完成后恢复上下文并返回，回到用户态。

4. (1)

寄存器-寄存器型：

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

寄存器-存储器型：

LOAD R1, B

ADD R1, C

STORE A, R1

ADD R1, C

STORE B, R1

ADD R1, A

STORE C, R1

累加器型：

LOAD B

ADD C

STORE A

ADD C

STORE B

ADD A

STORE C

堆栈型：

PUSH B

PUSH C

ADD

POP A

PUSH A

PUSH C

ADD

POP B

PUSH B

PUSH A

ADD

POP C

(2)

寄存器-存储器型：

指令长度均为x+y+z

总长度7x+7y+7z

内存位数7w

累加器型：

指令长度均为x+z

总长度7x+7z

内存位数7w

堆栈型：

PUSH/POP: x+z

ADD: x

总长度12x+9z

内存位数9w

寄存器-寄存器型：

LOAD/STORE: x+y+z

ADD: x+3y

总长度8x+14y+5z

内存位数5w

（3）寄存器-寄存器型更好。因为访问的内存位数更少，减少代码长度和访存时间。

5. 0xDEADBEEF

addr 0 1 2 3

Big-endian DE AD BE EF

Little-endian EF BE AD DE

6.

#include<stdio.h>

int main()

{

    int i;

    char x1 = (char)(-1);

    for(i = 1;(x1<<=1)!=0; i++);

    printf("char:%d\n",i);

    short x2 = (short)(-1);

    for(i = 1;(x2<<=1)!=0; i++);

    printf("short:%d\n",i);

    int x3 = (int)(-1);

    for(i = 1;(x3<<=1)!=0; i++);

    printf("int:%d\n",i);

    long x4 = (long)(-1);

    for(i = 1;(x4<<=1)!=0; i++);

    printf("long:%d\n",i);

    long long x5 = (long long)(-1);

    for(i = 1;(x5<<=1)!=0; i++);

    printf("long long:%d\n",i);

    return 0;

}

运行结果 (Core i7-8550u, VMware15, Ubuntu 18.04 64bit)

char:8

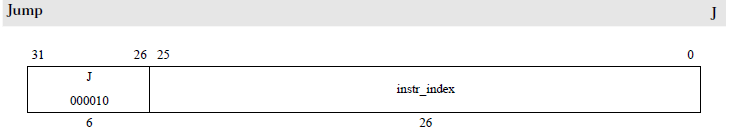
short:16

int:32

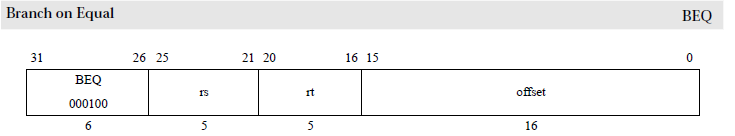
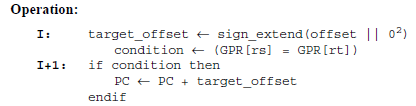
long:64

long long:64

7. MIPS32

直接跳转：

范围：最大为2^28，具体相对范围依赖于当前PC。

条件转移：

范围：-2^17~2^17-4

8.

LW R1, 0

SRL R1, R1, 8

LBU R2, 4

SLL R2,R2, 24

OR R1, R1, R2