第3次作业

1. 在x86-32 CPU中，逻辑地址由哪两部分组成？每个段与段寄存器之间有何对应的要求？
2. Intel CPU有哪六种寻址方式？各种寻址方式中，操作数的地址如何得到？
3. 保护方式的段寄存器的内容表示什么？它与实方式的段寄存器的内容有什么不同？
4. 在保护方式下，简述CPU如何获得用户程序A的内存单元ES:[12345H]的物理地址。
5. 在实模式下，编写一个程序段，将物理地址为 12345H的长字单元（4字节）的内容拷贝到EAX寄存器中。
6. 实方式下段内的第一个字节的偏移地址是多少？为什么实方式下每个段的起始物理地址一定能被16整除？
7. 在实模式下，分析下面的结果。

文本, 表格

中度可信度描述已自动生成

1. C程序中变量访问对应的寻址方式

int g;

void local\_variable\_visit()

{

int a[5];

int i, \*p;

g = 1;

a[0] = 10;

a[1] = 20;

i = 2;

a[i] = 30;

\*(a + 3) = 40;

p = &a[0];

p += 4;

\*p = 50;

printf("%d %d %d %d %d %d \n", g, a[0], a[1], a[2], a[3], a[4]);

}

观察反汇编代码，写出printf 语句之前，给g、a[0]、a[1]、a[2]、a[3]、a[4]赋值的汇编语句（赋值前进行的地址计算中间过程语句不需要写出来），指出它们分别使用的是什么寻址方式。指出g和a的地址表达形式。

1. 简述指令指示器EIP的作用。
2. 简述x86-32 CPU执行一条指令的过程（从取指机器码开始，直到指令执行完毕）。
3. 分析下面的C程序的运行结果，需要画出相关变量存贮图的变化过程。
4. **程序1**

#include <stdio.h>

int main()

{

short int i = 1;

long int j = 0x12345678;

short int \*p = &i;

\*p = 3;

printf(“%d”, i);

p = (short int \*)( (char \*)&j + 2 );

p[0] = 0x4142;

printf(“%x ”, j);

}

1. **程序2**

#include <stdio.h>

void main()

{

short int x = 1;

unsigned char \*p;

p = (unsigned char \*)&x;

p[0] = 0xFE;

p[1] = 0xFF;

printf(“%d”, x);

unsigned short \*q = &x;

\*q = 0x8000;

printf(“%d”, x);

}

1. **程序3**

#include <stdio.h>

union { long num;

struct { unsigned short n1;

unsigned short n2;

} b;

} a;

void main()

{

a.num = 0x12345678;

printf(“%x %x \n”, a.b.n1, a.b.n2);

a.num \*= -1;

printf(“%x %x \n”, a.b.n1, a.b.n2);

}