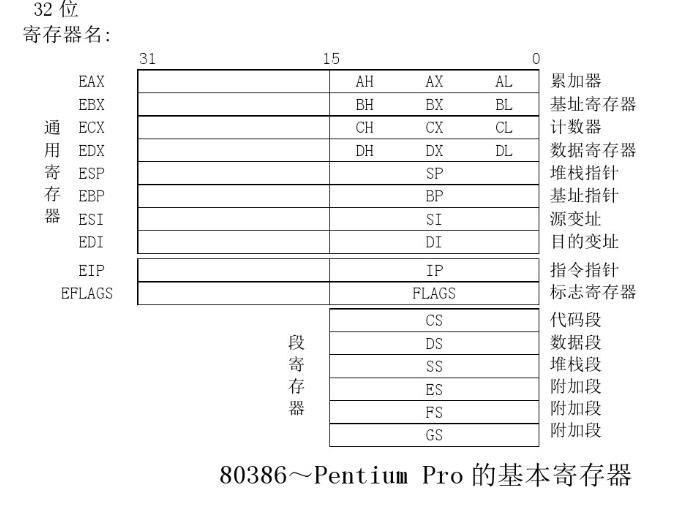
函数调用过程分析

栈：

1. 寄存器类型



64位的为R开头，例如RSP(堆栈指针)，RBP（堆栈基指针）【与栈有关】

Cpu包括通用寄存器和段寄存器

IP当前指令地址

**寄存器ebp称为“基址指针”，在未受改变之前始终指向栈底，用途是：在堆栈中寻址。**

**寄存器esp称为“栈指针”，会随着数据的入栈出栈移动，也就是说始终指向栈** **顶。**

例子【函数调用过程堆栈变化】

|  |
| --- |
| #include <stdio.h>  int add(int a , int b)  {  int c = a + b;  return c;  }  int main()  {  int result = 0;  result = add(1 , 2);  printf("%d\n",result);  return 0;  }  ---------------------  作者：z\_ryan  来源：CSDN  原文：https://blog.csdn.net/z\_ryan/article/details/81029195  版权声明：本文为博主原创文章，转载请附上博文链接！ |

|  |
| --- |
| pushq %rbp #rbp入栈 ，保存main的栈帧中的某个位置  movq %rsp, %rbp #ESP的内容赋值给了EBP。解放esp用于指向栈顶  movl $2, %esi #参数放入寄存器  movl $1, %edi  call add #调用add函数  movl %eax, -4(%rbp)  movl -4(%rbp), %eax  movl %eax, %esi  movl $.LC0, %edi  movl $0, %eax  call printf  movl $0, %eax  leave  ret |

|  |
| --- |
| Rbp |

|  |
| --- |
| pushq %rbp  movq %rsp, %rbp  movl %edi, -20(%rbp)  movl %esi, -24(%rbp)  movl -24(%rbp), %eax  movl -20(%rbp), %edx  addl %edx, %eax  movl %eax, -4(%rbp)  movl -4(%rbp), %eax  popq %rbp  ret |

