typedef常见用法  
  
1.常规变量类型定义  
  
例如：typedef unsigned char uchar  
描述：uchar等价于unsigned char类型定义 uchar c声明等于unsigned char c声明  
  
2.数组类型定义  
例如： typedef int array[2];  
描述： array等价于 int [2]定义; array a声明等价于int a[2]声明  
  
扩展： typedef int array[M][N];  
描述： array等价于 int [M][N]定义; array a声明等价于int a[M][N]声明  
  
3.指针类型定义  
例如： typedef int \*pointer;  
描述： pointer等价于 int \*定义;pointer p声明等价于int \*a声明  
  
例如： typedef int \*pointer[M];  
描述： pointer等价于 int \*[M]定义 pointer p声明等价于int \*a[M]声明明  
  
4.函数地址说明  
描述：C把函数名字当做函数的首地址来对待，我们可以使用最简单的方法得到函数地址  
例如： 函数:int func(void); unsigned long funcAddr=(unsigned long)func， funcAddr的值是func函数的首地址  
  
5.函数声明  
例如： typedef int func(void); func等价于 int (void)类型函数  
描述1： func f声明等价于 int f(void)声明，用于文件的函数声明  
描述2： func \*pf声明等价于 int (\*pf)(void)声明，用于函数指针的生命，见下一条  
  
6.函数指针  
例如： typedef int (\*func)(void)  
描述： func等价于int (\*)(void)类型  
func pf等价于int (\*pf)(void)声明，pf是一个函数指针变量  
  
7.识别typedef的方法：  
a).第一步。使用已知的类型定义替代typdef后面的名称,直到只剩下一个名字不识别为正确  
如typedef u32 (\*func)(u8);  
从上面的定义中找到 typedef \_\_u32 u32;typedef \_\_u8 u8  
继续找到 typedef unsigned int \_\_u32;typedef unsigned char \_\_u8;  
替代位置名称 typedef unsigned int (\*func)(void);  
现在只有func属于未知。  
b).第二步.未知名字为定义类型，类型为取出名称和typedef的所有部分，如上为  
func等价于unsigned unsigned int (\*)(unsigned char);  
c).第三部.定义一个变量时，变量类型等价于把变量替代未知名字的位置所得到的类型  
func f等价于unsigned unsigned int (\*f)(unsigned char)