快速平方根算法

求平方根的代码

虽然这两篇文章将的都是同一个函数的同种实现方法，但是由于写法有一点点差异

前一片中的方法测出的结果和实际结果完全两码事，而下面这种写法就没有什么问题，

我已经在 VC2005 和 Gcc4.1 上测试 OK，前一篇中的写法可能被编译器误解优化导致

结果出现问题。

```
/*================SquareRootFloat================*/
float SquareRootFloat(float number)
{
long i;
float x, y;
const float f = 1.5F;
       x = number * 0.5F;
       y = number;
       i = * ( long * ) &y;
       i = 0x5f3759df - ( i >> 1 ); //注意这一行
       y = * ( float * ) &i;
       y = y * ( f - ( x * y * y ) );
       y = y * ( f - ( x * y * y ) );
return number * y;
}
```

0x5f3759df 是什么？简单来说比如求 5 的平方根，选一个猜测值比如 2，那么可以这么算
5/2 = 2.5; 2.5+2/2 = 2.25; 5/2.25 = xxx; 2.25+xxx/2 = xxxx ...

这 样反复迭代下去，结果必定收敛于 sqrt(5)，一般的求平方根都是这么算的。而卡马克的不同之处在于，他选择了一个神秘的猜测值 0x5f3759df 作为起始，使得整个逼近过程收敛速度很快，对于 Quake III 所要求的精度 10 的负三次方，只需要一次迭代就能够得到结果。
普渡大学的数学家 Chris Lomont 看了以后觉得有趣，决定要研究一下卡马克弄

出来的这个猜测值有什么奥秘。Lomont 在精心研究之后从理论上也推导出一个最佳猜测值，和卡马克的数字非常接近，0x5f37642f。

传奇并没有在这里结束。Lomont 计算出结果以后非常满意，于是拿自己计算出的起始值和卡马克的神秘数字做比赛，看看谁的数字能够更快更精确的求得平方根。结果是卡马克赢了... 谁也不知道卡马克是怎么找到这个数字的。
最后 Lomont 采用暴力方法一个数字一个数字试过来，终于找到一个比卡马克数字要好上那么一丁点的数字，虽然实际上这两个数字所产生的结果非常近似，这个暴力得出的数字是 0x5f375a86。
Lomont 为此写下一篇论文，"Fast Inverse Square Root"。