Subject: Re: "better" version of Iscale functions
Posted by mdelfede on Sun, 06 Apr 2008 18:05:38 GMT
View Forum Message <> Reply to Message

Btw, I guess GCC does some better optimizazions in this case :

```
int iscale(int x, int y, int z)
{
  int64_t res = x;
  res *= y;
  res /= z;
  return (int)res;
}
```

gets translated as :

```
.globl _ZN3Upp6iscaleEiii
 .type _ZN3Upp6iscaleEiii, @function
_ZN3Upp6iscaleEiii:
.LFB4039:
 .file 4 "/home/massimo/sources/upp-svn/uppsrc/Core/mathutil.cpp"
 .loc 4 11 0
 pushq %rbp
.LCFI15:
 movq %rsp, %rbp
.LCFI16:
 movl %edi, -20(%rbp)
 movl %esi, -24(%rbp)
 movl %edx, -28(%rbp)
.LBB2:
 .loc 4 17 0
 movl -20(%rbp), %eax
 cltq
 movq %rax, -8(%rbp)
 .loc 4 18 0
 movl -24(%rbp), %eax
 movslq %eax,%rdx
 movq -8(%rbp), %rax
 imulq %rdx, %rax
 movq %rax, -8(%rbp)
 .loc 4 19 0
 movl -28(%rbp), %eax
 cltq
 movq -8(%rbp), %rdx
 movq %rax, %rcx
```

```
 movq %rdx, %rax
 sarq $63, %rdx
 idivq %rcx
 movq %rax, -8(%rbp)
 .loc 4 20 0
 movq -8(%rbp), %rax
.LBE2:
 .loc 4 30 0
 leave
 ret
```

If you drop assembly directive and/or line references and other stuffs, you can see that work is done with just one imul and one idiv... like your assembly version. The rest is just registry moving stuffs. I don't know what happens if you compile it with full optimization, but I guess mostly of them just disappears.
And I think that's still much quicker than floating point version.

Max
