
Subject: Re: SSE2(/AVX) and alignment issues
Posted by [Novo](#) on Fri, 28 Jan 2011 20:53:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Fri, 28 January 2011 04:55: However, one big problem is that you would not be able to use 'new' to allocate SSE2 objects.

Of course, there is not much use for 'new' in U++ anyway, but still...

What is SSE2 object?

You can allocate `sizeof(object) + alignment - 1` of memory and use placement new/delete to call constructor/destructor.

```
// A must be power of two.
template <unsigned A>
void* AlignMem(void* addr)
{
    unsigned a = A - 1;
    return (void*)((unsigned)addr + a) & ~a;
}

void* ptr = UPP::MemoryAlloc(sizeof(object) + alignment - 1)
object* o = new(AlignMem<alignment>(ptr)) object(aarg1, arg2, e.t.c.);
...
o->~object();
UPP::MemoryFree(ptr);
```

You need an extra pointer to allocated memory.

As far as I understand these SSE2 commands are going to be used with complicated data structures like matrices, vectors, strings, e.t.c. So, this pointer and allocation/destruction logic can be hidden inside of overloaded new/delete.

I personally would prefer to have an allocator, which allocates memory aligned by 16/32/64 bytes. AVX seems to be supported by Intel only at this time. So, you do not want to use AVX for quite long time. By the time we will need 64-byte aligned memory it might cost 1\$ for a Gbyte.

mirek wrote on Fri, 28 January 2011 04:55
(Actually, life would be much easier in C++ if there would not be any 'new' and 'delete')

Mirek

Garbage collector in C++ is a real pain in the neck also. Your object can be garbage collected

when you are still in a constructor.

I personally would prefer new/delete.
