
Subject: Re: SSE2(/AVX) and alignment issues
Posted by [mirek](#) on Fri, 28 Jan 2011 23:03:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Fri, 28 January 2011 15:53mirek wrote on Fri, 28 January 2011 04:55However, 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.

This is not a question. The question is whether `_regular_` 'new' should return 16-byte aligned values or not. (And later, with AVX, 32, then maybe in 4 more years 64 etc...)

As long as we agree that allocating SSE2 stuff with 'new' is not a regular thing, we are at option 2..

Mirek