Subject: Optimized memcmp for x86
Posted by mirek on Fri, 22 Feb 2008 10:07:16 GMT
View Forum Message <> Reply to Message

Well, this code seems to run 20% faster than intrinsic GCC memcmp on x86-64:

```
#ifdef COMPILER_GCC
inline dword _byteswap_ulong(dword x)
{
 asm("bswap %0" : "=r" (x) : "0" (x));
 return x;
}

inline uint64 _byteswap_uint64(uint64 x)
{
 asm("bswap %0" : "=r" (x) : "0" (x));
 return x;
}

inline word _byteswap_ushort(word x)
{
 __asm__("xchgb %b0,%h0" : "=q" (x) :  "0" (x));
 return x;
}
#endif

int MemCmp(const char *a, const char *b, size_t len)
{
 if(((size_t)a & 3) | ((size_t)b & 3))
  return memcmp(a, b, len);
 const dword *x = (dword *)a;
 const dword *y = (dword *)b;
 const dword *e = x + (len >> 2);
 while(x < e) {
  if(*x != *y)
   return int(_byteswap_ulong(*x) - _byteswap_ulong(*y));
  x++;
  y++;
 }
 if(len & 2)
  if(*(word *)x != *(word *)y)
   return int(_byteswap_ushort(*(word *)x) - _byteswap_ushort(*(word *)y));
 if(len & 1)
  return int(*((byte *)x + 2)) - int(*((byte *)y + 2));
 return 0;
}
```

(Obviously, when both areas are dword aligned, but that happens a lot...).

Mirek
