Subject: Re: BufferPainter::Clear() optimization
Posted by Tom1 on Wed, 20 May 2020 17:51:10 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 20 May 2020 18:31I am getting quite different picture:

```
 int bsize=8*1024*1024;
 Buffer<dword> b(bsize, 0);
 dword cw = 123;

 String result="\"N\",\"memsetd()\",\"Fill3T3()\"\r\n";
 for(int len=1;len<=bsize;){
  int maximum=100000000/len;
  int64 t0=usecs();
  for(int i = 0; i < maximum; i++)
   memsetd(~b, cw, len);
  int64 t1=usecs();
  for(int i = 0; i < maximum; i++)
   Fill3T3(~b, cw, len);
  int64 t2=usecs();
  String r = Format("%d,%f,%f",len,1000.0*(t1-t0)/maximum,1000.0*(t2-t1)/maximum);
  RLOG(r);
  result.Cat(r + "\r\n");
  if(len<64) len++;
  else len*=2;
 }

 SaveFile(GetHomeDirFile("memset.csv"),result);
```

I am starting to wonder if there is difference between our MSC 32bit compilers...

Hi,

No wonder we ended up with (very slightly) different approach... Your results are more or less reversed to what I'm getting. I tried to reorder the calls too, but without any observable difference.

It's either the different CPUs or a different compiler. My compiler is:

Microsoft (R) C/C++ Optimizing Compiler Version 19.21.27702.2 for x86
Copyright (C) Microsoft Corporation. All rights reserved.

Should I downgrade or upgrade?...

Anyway, seriously I'm pleased with the final result here. The filler is now better than anything before and can be used generally for all clearing/presetting of buffers. I use this a lot in signal processing in addition to clearing the ImageBuffer for BufferPainter. After all, the ImageBuffer

needs to be cleared or preset to user preference background color once before each display update. It is much better to have a 1.5 ms delay instead of 3.6 ms delay before drawing approximately 10-20 ms worth of vector map data on the screen. :)

Should this new memsetd() now be deployed all over the u++? I mean e.g. Core/Topt.h :: Fill?

Thank you a lot for your great work on this! :)

Best regards,

Tom
