gtest集成内存测试工具memtester

1> 原理

主要是捕获内存错误和一直处于很高或很低的坏位，其测试的主要项目是随机值、异或比较、减法、乘法、除法、与或运算等等，通过给定测试内存的大小和次数，可以对系统现有的内存进行上面项目的测试；

struct test tests[] = {

    { "Random Value", test\_random\_value },

    { "Compare XOR", test\_xor\_comparison },

    { "Compare SUB", test\_sub\_comparison },

    { "Compare MUL", test\_mul\_comparison },

    { "Compare DIV",test\_div\_comparison },

    { "Compare OR", test\_or\_comparison },

    { "Compare AND", test\_and\_comparison },

    { "Sequential Increment", test\_seqinc\_comparison },

    { "Solid Bits", test\_solidbits\_comparison },

    { "Block Sequential", test\_blockseq\_comparison },

    { "Checkerboard", test\_checkerboard\_comparison },

    { "Bit Spread", test\_bitspread\_comparison },

    { "Bit Flip", test\_bitflip\_comparison },

    { "Walking Ones", test\_walkbits1\_comparison },

    { "Walking Zeroes", test\_walkbits0\_comparison },

#ifdef TEST\_NARROW\_WRITES

    { "8-bit Writes", test\_8bit\_wide\_random },

    { "16-bit Writes", test\_16bit\_wide\_random },

#endif

    { NULL, NULL }

};

2> 获取剩余内存的大小

测试之前可以查看一下当前剩余内存还有多少，便于进行接下来的测试；

3> 简单测试

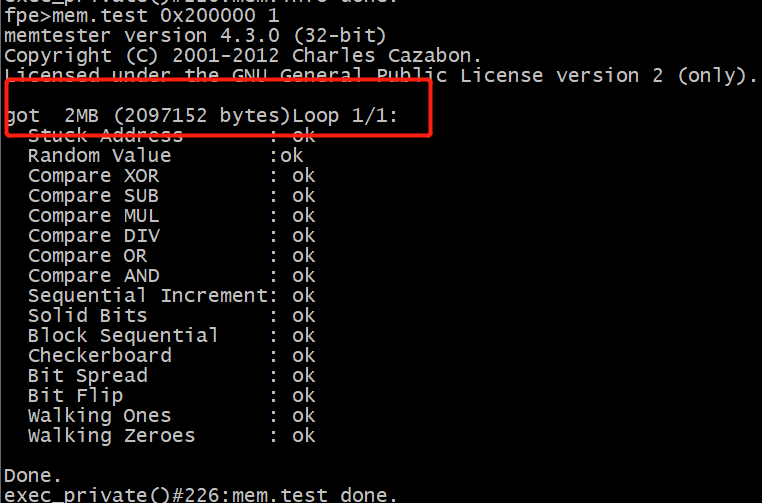
REGISTER\_CMD(mem\_test, "测试内存操作", "mem\_size|loops(0为一直循环)",  "0x200000 1");

mem\_size: 需要测试的内存大小

loops：测试次数

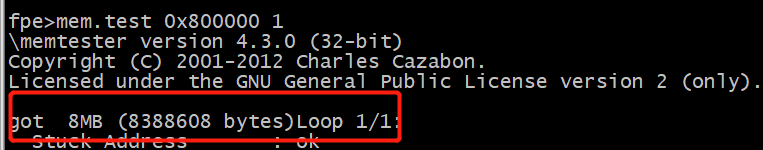
//测试2M内存大小

mem\_test 0x200000 1



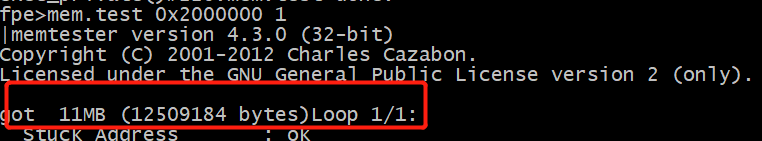
// 说明当前可用内存>2M

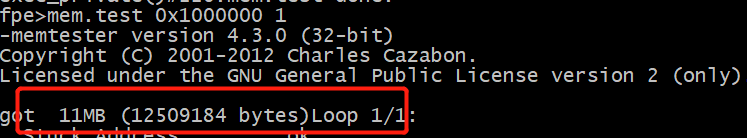
申请8M内存：

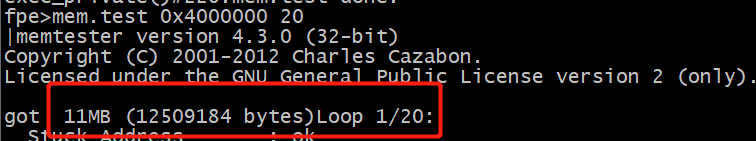


//说明当前可用内存>8M

申请16M、32M、64M（DDR为64M），显示可用内存只用11M了







3> 正式测试11M内存，循环测试20次，差不多用时30min（剩余的最大内存）

