# 程序运行说明

**运行环境：**

服务器：

* openEuler 20.03 (LTS), Linux version 4.19.90-2003.4.0.0036.oe1.aarch64
* gcc (GCC) 7.3.0

虚拟机：

* openEuler 20.03 (LTS), Linux version 4.19.90-2003.4.0.0036.oe1.x86\_64
* gcc (GCC) 7.3.0

**程序输入：**

通过读取同文件夹下 input.txt 文件来读取内存操作，该文件中的一行表示一个操作

0 0 50 7

1 0 40 7

0 50 40 7

1 50 30 7

2 50 20 7

3 50 10 7

4 50 20 7

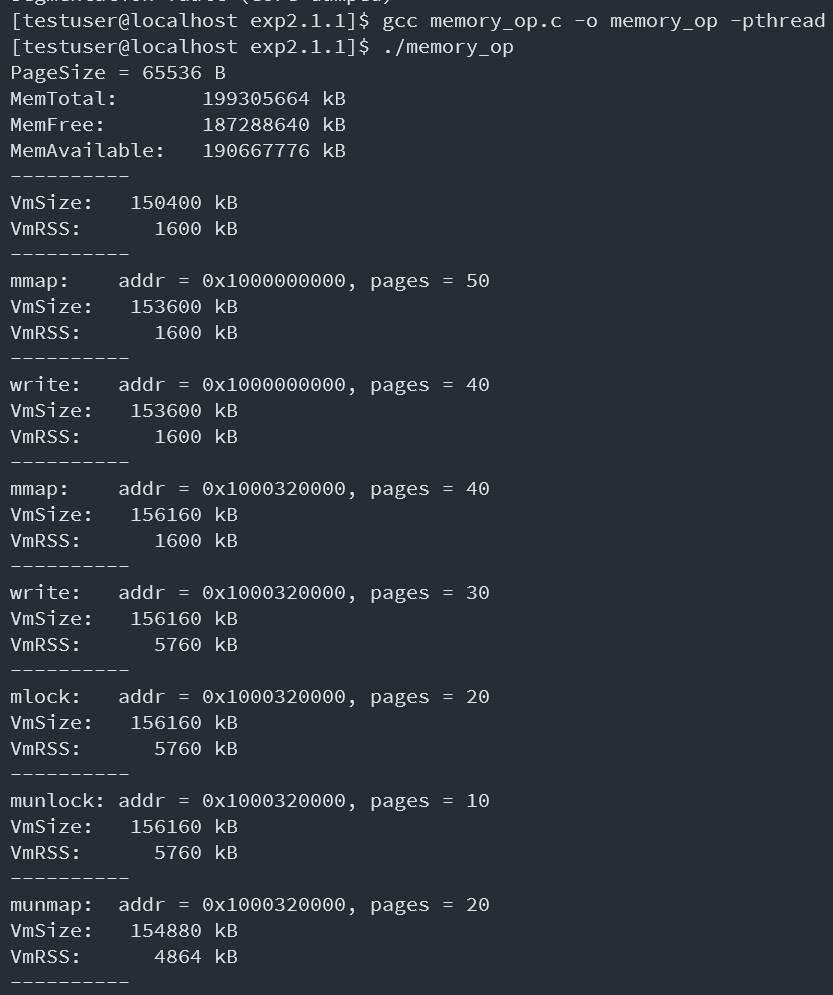
第 1列是内存操作代号（0: mmap, 1: write, 2: mlock, 3: munlock, 4: munmap），第 2 列是操作的相对起始页码，第 3列是操作的页数，第4列是权限位串的十进制值（如7=0111代表 PROT*READ|PROT*WRITE|PROT\_EXEC）。

**编译：**

gcc memory\_op.c -o memory\_op -l pthread

**运行：**

./memory\_op



输出的意义：

PageSize = 65536 B //页面大小为 65536B，即64KB

MemTotal: 199305664 kB //系统总内存

MemFree: 187288640 kB //系统空闲内存

MemAvailable: 190667776 kB //系统可用内存

----------

VmSize: 150400 kB //线程t1使用的虚拟内存

VmRSS: 1600 kB //线程t2使用的物理内存

----------

mmap: addr = 0x1000000000, pages = 50 //分配50页虚拟内存

VmSize: 153600 kB //t1使用的虚拟内存增加了50\*64KB

VmRSS: 1600 kB //t1使用的物理内存未增加

----------

write: addr = 0x1000000000, pages = 40 //弄脏40页

VmSize: 153600 kB

VmRSS: 1600 kB

----------

mmap: addr = 0x1000320000, pages = 40 //分配40页

VmSize: 156160 kB //t1使用的虚拟内存增加40\*64KB

VmRSS: 1600 kB

----------

write: addr = 0x1000320000, pages = 30 //弄脏30页

VmSize: 156160 kB

VmRSS: 5760 kB //p1使用的物理内存增加4160KB（65页）

----------

mlock: addr = 0x1000320000, pages = 20 //锁定20页

VmSize: 156160 kB

VmRSS: 5760 kB

----------

munlock: addr = 0x1000320000, pages = 10 //解锁20页

VmSize: 156160 kB

VmRSS: 5760 kB

----------

munmap: addr = 0x1000320000, pages = 20 //释放20页

VmSize: 154880 kB //t1使用的虚拟内存减少20页

VmRSS: 4864 kB //t1使用的物理内存减少14页

----------

可以看到，分配 n 页虚拟内存后，虚拟内存增加 n 页，物理内存不增加；写 n 页后，物理内存增加不定。释放 n 页后，虚拟内存减少 n 页，物理内存减少不定。