华东师范大学数据学院上机实践报告

课程名称：操作系统　　**年级：**2019 级　　**上机实践成绩：**

指导教师：翁楚良　　**姓名：**叶秋雨

上机实践名称：I/O subsystem　　**学号：**10184102103　　**上机实践日期：**

上机实践编号：

一、 目的

1、熟悉类 UNIX 系统的 I/O 设备管理

2、熟悉 MINIX 块设备驱动

3、熟悉 MINIX RAM 盘

二、 内容与设计思想

• 在 Minix3 中安装一块 x MB 大小的 RAM 盘，可以挂载并且存取文件操作。

• 编写测试文件，测试不同块大小下、不同块扫描方式（顺序/随机）RAM 盘和 Disk 盘的文件读写速度并分析其读写速度差异原因。

三、 使用环境

物理机：Windows10

虚拟机：Minix3

虚拟机软件：Vmware

终端控制软件：MobaXterm

物理机与虚拟机文件传输：FileZilla

四、 实验过程

（1）在 Minix3 中安装一块 RAM 盘

• 修改/usr/src/minix/drivers/storage/memory/memory.c，增加默认的用户 RAM 盘数。重新编译内核，重启虚拟机。

```
/* ramdisks (/dev/ram*) */
#define RAMDISKS     7
```

• 创建设备 mknod /dev/myram b 1 13，并输入 ls /dev/ | grep ram 查看设备是否创建成功。

```
# ls /dev/ | grep ram
myram
ram
ram0
ram1
ram2
ram3
ram4
ram5
#
```

• 参考/usr/src/minix/commands/ramdisk/ramdisk.c，实现 buildmyram.c 容量初始化工具，将单位从 KB 修改为 MB。同时，在同一目录下的 Makefile 文件中添加相应条目。重新编译内核，重启虚拟机。

```
#define KFACTOR 1048576
```

```
PROG=   ramdisk
PROG=   buildmyram
```

• 执行命令 buildmyram <size in MB> /dev/myram，创建一个 RAM 盘。

```
# buildmyram 500 /dev/myram
size on /dev/myram set to 500MB
#
```

• 在 RAM 盘上创建内存文件系统，mkfs.mfs /dev/myram。

• 将 RAM 盘挂载到用户目录下，mount /dev/myram /root/myram，输入 df 查看是否挂载成功。

```
# mount /dev/myram /root/myram
/dev/myram is mounted on /root/myram
# df
Filesystem    512-blocks       Used      Avail %Cap Mounted on
/dev/myram       1024000      16088    1007912   1% /root/myram
/dev/c0d3p0s0     262144     262144          0 100% /
none                   0          0          0 100% /proc
/dev/c0d3p0s2   33566464   11188784   22377680  33% /usr
/dev/c0d3p0s1    8114176      84968    8029208   1% /home
none                   0          0          0 100% /sys
#
```

注意：重启后用户自定义的 RAM 盘内容会丢失，因此每次重启后都需要重新设置大小，创建文件系统并挂载。

（2）编写测试代码

• 主函数实现思路：

分别调用相应函数测试在顺序和随机两种块扫描方式下，RAM 盘和 Disk 盘多个进程并发读写不同大小块的时间，并计算平均读写速度。

-为每个进程分配独立的文件

-wait(NULL)函数等待所有子进程读写结束再记录结束时间

-经测试，进程并发度在 7~10 之间吞吐量达到最大，基于此在实验中可以修改并发度大小来获得良好的实验数据

```
for(int blocksize=64;blocksize<=1024*32;blocksize=blocksize*2){
            int Concurrency=7;
            gettimeofday(&starttime, NULL);
            for(int i=0;i<Concurrency;i++){
                if(fork()==0){
                //随机写
                //write_file(blocksize,true,filepathDisk[i]);
                //write_file(blocksize,true,filepathRam[i]);

                //顺序写
                //write_file(blocksize,false,filepathDisk[i]);
                //rite_file(blocksize,false,filepathRam[i]);

                //随机读
                read_file(blocksize,true,filepathDisk[i]);
                //read_file(blocksize,true,filepathRam[i]);

                //顺序读
                //read_file(blocksize,false,filepathDisk[i]);
                //read_file(blocksize,false,filepathRam[i]);
                exit(0);
                }
            }
            //等待所有子进程结束
            //wait 失败返回-1
            while(wait(NULL)!=-1);
            gettimeofday(&endtime, NULL);
            spendtime=get_time_left(starttime,endtime)/1000.0;//换算成秒
            int block=blocksize*Concurrency*times;
            printf("blocksize_KB=%.4fKB,speed=%fMB/s\n",(double)blocksize/1024.0,(d
ouble)block/spendtime/1024.0/1024.0);
    }
```

• write_file & read_file 函数实现思路：

通过多次重复的读写操作来计算 RAM 盘/Disk 的读写速度。

-若为随机读写，则每次读写结束后利用 lseek(fp,rand()%(filesize-blocksize),SEEK_SET)函数定位到文件任意位置。

-为了减小主机操作系统的缓存机制造成的误差，将文件大小 filesize 设置为 300MB

注意：对(filesize-blocksize)取余，防止写出文件

```
//blocksize 表示写的块大小
//israng=true 表示随机写，否则为顺序写
//filepath 为文件写的路径
void write_file(int blocksize, bool israng, char *filepath){
    int fp=open(filepath,O_RDWR|O_CREAT|O_SYNC,0755);
    if(fp==-1) printf("open file error!\n");
    int res;
    //多次重复写入计算时间
    for(int i=0;i<times;i++){
        if((res=write(fp,buff,blocksize))!=blocksize){
            printf("%d\n",res);
            printf("write file error!\n");
        }
        if(israng){
            //随机生成一个整数 取余定位到文件中任意位置
            //对（filesize-blocksize）取余，防止写出文件
            lseek(fp,rand()%(filesize-blocksize),SEEK_SET);
        }
    }
    lseek(fp,0,SEEK_SET);
}

void read_file(int blocksize,bool israng,char *filepath){
    int fp=open(filepath,O_RDWR|O_CREAT|O_SYNC,0755);
    if(fp==-1) printf("open file error!\n");
    int res;
    for(int i=0;i<times;i++){
        if((res=read(fp,readbuff,blocksize))!=blocksize){
            printf("%d\n",res);
            printf("read file error!\n");
        }
        if(israng){
            lseek(fp,rand()%(filesize-blocksize),SEEK_SET);
        }
    }
    lseek(fp,0,SEEK_SET);
}
```

• get_time_left 函数实现思路：

利用测试得到的 starttime 和 endtime 计算读写所耗费的时间。

```
long get_time_left(struct timeval starttime,struct timeval endtime){
    long spendtime;
```

```
    //换算成毫秒
    spendtime=(long)(endtime.tv_sec-starttime.tv_sec)*1000+(endtime.tv_usec-
starttime.tv_usec)/1000;
    return spendtime;
}
```

五、实验结果

（1）实验数据表

随机写	64B	128B	256B	512B	1KB	2KB	4KB	8KB	16KB	32KB
磁盘	0.2441	3.2552	5.0862	6.1035	8.138	10.6524	12.0787	14.4222	15.4519	26.0417
RAM 盘	2.6308	6.1035	9.7657	21.0466	42.0932	73.4258	97.6563	167.6502	233.9072	260.4167

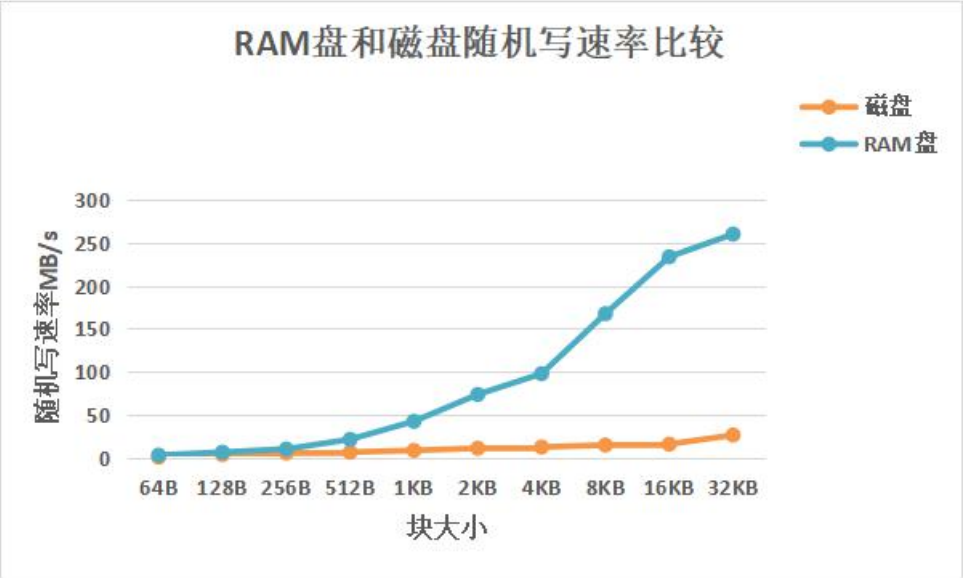
顺序写	64B	128B	256B	512B	1KB	2KB	4KB	8KB	16KB	32KB
磁盘	2.5894	5.0264	10.3575	20.715	32.9443	41.4299	131.7771	164.4737	170.8984	201.6129
RAM 盘	3.5903	7.398	12.207	29.5928	39.0625	78.125	156.25	188.253	234.9624	454.5455

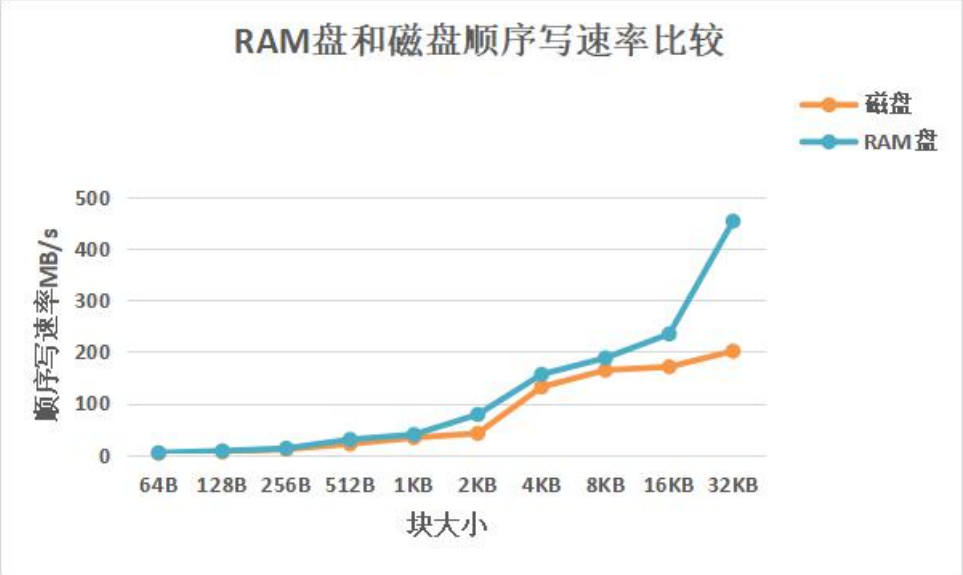
随机读	64B	128B	256B	512B	1KB	2KB	4KB	8KB	16KB	32KB
磁盘	1.7089	3.418	6.8359	10.3575	16.4721	23.5721	32.9443	43.75	45.2898	56.9661
RAM 盘	1.8384	4.069	7.3536	32.552	58.8291	83.4669	180.8449	267.5514	312.5	374.7002

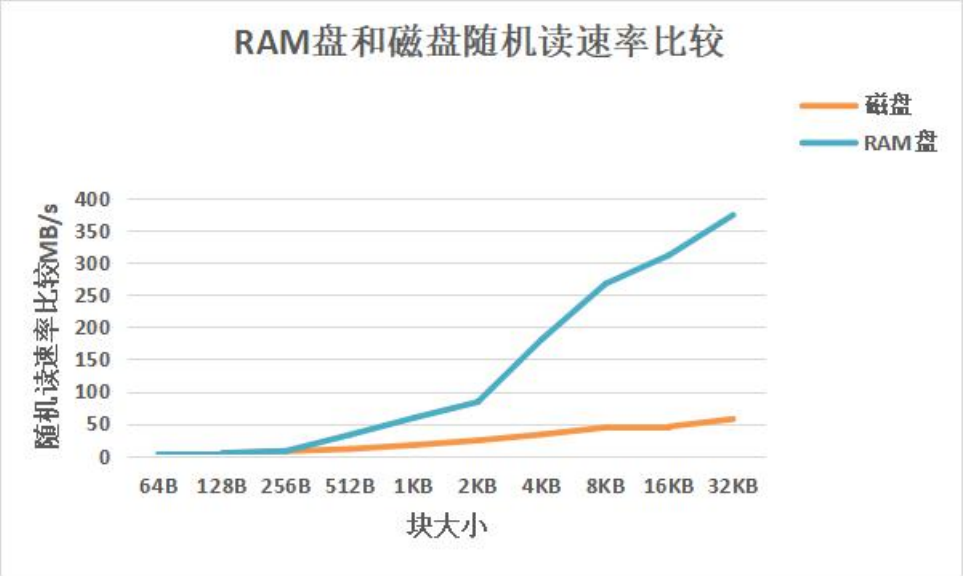
顺序读	64B	128B	256B	512B	1KB	2KB	4KB	8KB	16KB	32KB
磁盘	2.059	6.8359	19.5313	39.0625	59.1856	156.25	165.7197	260.4167	312.5	437.5
RAM 盘	5.0265	10.3575	20.715	42.7446	82.8598	165.7197	218.75	331.4394	437.5	527.1084

注：读写速率的单位：MB/s

（2）实验数据图




RAM盘和磁盘顺序写速率比较
磁盘
RAM盘
顺序写速率MB/s
500
400
300
200
100
0
64B 128B 256B 512B 1KB 2KB 4KB 8KB 16KB 32KB
块大小


RAM盘和磁盘随机读速率比较
磁盘
RAM盘
随机读速率比较MB/s
400
350
300
250
200
150
100
50
0
64B 128B 256B 512B 1KB 2KB 4KB 8KB 16KB 32KB
块大小


RAM盘和磁盘顺序读速率比较
磁盘
RAM盘
顺序读速率MB/s
600
500
400
300
200
100
0
64B 128B 256B 512B 1KB 2KB 4KB 8KB 16KB 32KB
块大小

（3）实验结果分析

总体来看，写的速率比读的速率慢，随机读写的速率比顺序读写的速率慢，尤其是磁盘两种快扫描方式的速率差异更加明显。究其原因在于磁盘读写时，机械寻道是影响磁盘读写速率的主要因素，随机读写每次都需要重新寻道，而顺序读写只需要在第一次进行寻道，因此两种方式的读写速率差别很大。

对比 Disk 盘和 RAM 盘，RAM 盘的读写速率普遍快于 Disk 盘，特别是随机读写性能。由于 RAM 盘为内存分配的一块区域，因此没有寻道和旋转延迟，所以 RAM 的读写速率优于 Disk。但对 Disk 进行一次读后，其内容会被缓存，之后再次读 Disk 上的数据，其效率有所提高。

六、总结

通过本次实验我更加熟悉了类 UNIX 系统的 I/O 设备管理，熟悉了 MINIX 块设备驱动，熟悉了 MINIX RAM 盘。同时体会到了 RAM 盘和 Disk 读写速度的差异，更好地理解了教材的内容。

在实验过程中，也遇到了一些问题。比如随机读写在文件中的重新定位靠考虑超出文件大小的问题；在实验数据不理想的情况下，要及时调整参数，如：读写次数、进程并发数等等来获得好的实验数据；对磁盘的测试，每次要重启虚拟机，尽量减少缓存对读写速率的影响。