

《操作系统》实验

三、I/O subsystem

助教：彭小双、于春琳

指导教师：翁楚良

2022 春 ECNU

目标

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

实验要求

- 在MINIX3中安装一块X MB大小的RAM盘（minix中已有6块用户可用RAM盘，7块系统保留RAM盘），可以挂载并且存取文件操作。
- 测试RAM盘和DISK盘的文件读写速度，分析其读写速度差异原因（可用图表形式体现在实验报告中）。

实现说明

- 增加RAM盘：
- 修改/usr/src/minix/drivers/storage/memory/memory.c，增加默认的用户RAM盘数：RAMDISKS=7。
- 重新编译内核，重启reboot。
- 创建设备mknod /dev/myram b 1 13，查看设备是否创建成功输入 ls /dev/ | grep ram。
- 实现buildmyram初始化工具（用于分配容量）。
 - 参考/usr/src/minix/commands/ramdisk/ramdisk.c，实现buildmyram.c，但是需要将KB单位修改成MB。
 - 编译buildmyram.c文件，然后执行命令：buildmyram <size in MB> /dev/myram。创建一个RAM盘。

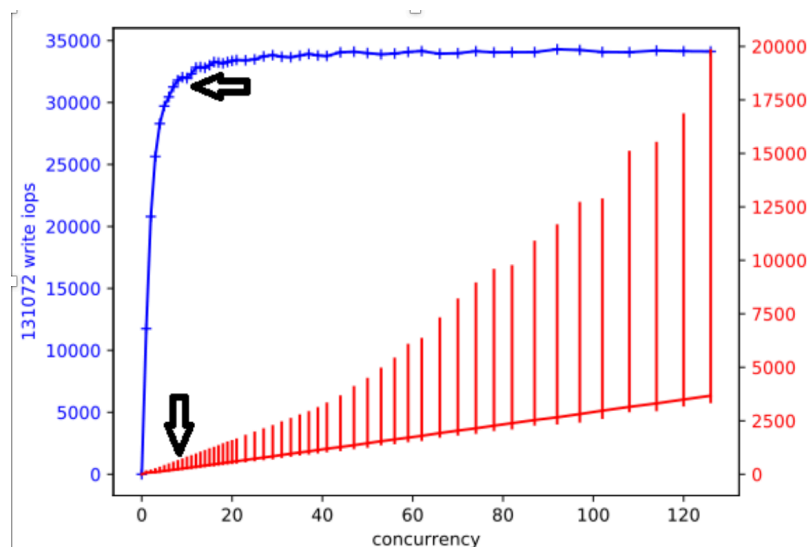
实现说明(续)

- 增加RAM盘：
- 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`,查看是否挂在成功：输入`df`。
- 注：重启后用户自定义的ram盘内容会丢失，需要重新设置大小，创建文件系统，并挂载。

实现说明(续)

■ 性能测试：

1. RAM盘和Disk盘的性能测试中，需要采用多进程并发的同步读写，并发数要增加到设备接近“饱和”状态（吞吐量难以继续提升，但是I/O延时恶化）。在出现饱和前，总吞吐量随着并发数线性增长。



计算公式：总吞吐量=总文件大小/执行时间

实现说明(续)

■ 性能测试：

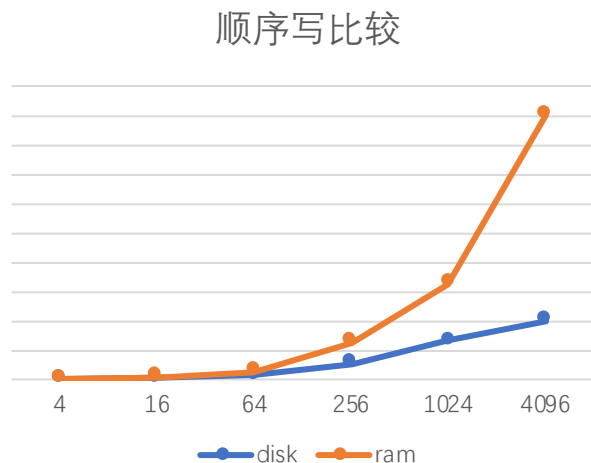
2. 性能测试的二个变量为“块大小”（推荐64B/256B/1KB/4KB/16KB/64KB）和“块扫描方式”（顺序/随机）。可以画四张曲线图对比RAM盘和Disk盘性能（随机读，随机写，顺序读，顺序写）。实验结果预计为RAM盘性能高于DISK盘，特别是随机读写性能。

注意事项

- 使用posix函数 open打开文件，利用O_SYNC参数使得write/read操作为同步模式。
- 一定要检查write/read函数的返回值，以及写入的字节数目，确定是否成功。
- 为了简化实验，可以为每个进程分配一个独立的文件。为了减小主机操作系统的缓存机制造成的误差，文件总大小越大越好（例如300MB）。
- 随机读写时，可以采用lseek重新定位文件指针；顺序读写时，默认文件指针自动移动，当读到文件末尾时，可以用lseek返回文件头。
- 每组的读写需要反复持续一段时间，过短的时间会造成误差较大。
- 通常情况下，7~15个进程达到饱和，吞吐量不会高于700MB/s (ram盘顺序读写)。
- 如果minix虚拟机建在SSD下，会导致随机和顺序的差距减小，所以最好把虚拟机放在机械硬盘上，实验效果更明显。

测试用例

- 分别测试在RAM盘和disk盘下的随机读，顺序读，随机写，顺序写速度。对比在不同的blocksize下的性能，然后通过Excel把数据制作成图表。



```
/*写文件:打开文件,判断返回值,如果正常打开文件就判断是否随机写,进行写操作*/
void write_file(int blocksize, bool isrand, char *filepath, int fs)
{
    //to do...
}

/*读文件:打开文件,判断返回值,如果正常打开文件就判断是否随机读,进行读操作*/
void read_file(int blocksize, bool isrand, char *filepath, int fs)
{
    //to do...
}

//计算时间差,在读或写操作前后分别取系统时间,然后计算差值即为时间差。
long get_time_left(long starttime, long endtime)
{
    //to do...
    return spendtime;
}

/*主函数: 首先创建和命名文件,通过循环执行read_file和write_file函数测试读写差异。
测试blocksize和concurrency对测试读写速度的影响,最后输出结果。*/
int main()
{
    //随机写
    //write_file(blocksize, true, &filepath, filesize/concurrency);
    //顺序写
    //write_file(blocksize, false, &filepath, filesize/concurrency);
    //随机读
    //read_file(blocksize, true, &filepath, filesize / concurrency);
    //顺序读
    //read_file(blocksize, false, &filepath, filesize / concurrency);

    /*等待子进程完成后,获取计算时间,计算读写操作所花时间,延时,吞吐量等*/
    //to do...
    return 0;
}
```

实验提交

1. 提交材料：

1. 该project的实验报告，测试代码，Excel数据图表。
2. 相关源代码修改增量，内核补丁git diff > change.txt。

2. 实验要求：

1. 独立完成实验，严禁抄袭。
2. 请在规定时间内提交实验。