《操作系统原理》实验报告

年级：2019 级　班级：大类 8 班　学号：3019244266　姓名：李润泽

一、实验题目及要求

实验题目：LAB4-bigger file for xv6

实验要求：1、阅读 xv6 源代码，理解 inode 结构体与 dinode 结构体的作用，里面重要字段的作用。

2、xv6 文件系统原本有 13 个索引项，其中 12 个直接索引，一个一级间接索引，请计算出 xv6 系统能支持的最大文件大小，并在实验报告中描述计算过程。

3、若索引项总个数不变，修改 xv6 索引项，减少一个直接索引，增加一个二级间接索引，从而增大 xv6 能支持的文件大小。经过这样的修改，能支持的最大文件大小是多少？请在实验报告中描述计算过程。

4、修改内核源代码 fs.c，减少一个直接索引，增加一个二级间接索引，从而增大 xv6 能支持的文件大小。

5、编写用户程序 big.c，验证修改后的内核正确性。

二、设计说明(用来说明程序的功能、结构、原理等)

首先修改 Makefile，将 CPUS := 2 修改为 CPUS := 1，并在 QEMUOPTS 前添加 QEMUEXTRA = -snapshot

修改 param.h 中#define FSSIZE 1000 为#define FSSIZE 20000

在目录中添加 big.c 文件（该文件在作业文档中已提供），并在 Makefile 文件的 UPROGS 中添加一行 _big\

修改 fs.c 文件中的 bmap()函数以及 fs.h 中的定义，使其减少一个直接索引，增加一个二级间接索引。

修改代码如下图：

fs.c

```
394       log_write(bp);
395     }
396     brelse(bp);
397     return addr;
398   }
399   bn -= NINDIRECT;
400
401   if(bn < NINDIRECT*NINDIRECT){
402     // Load first indirect block, allocating if necessary
403     if((addr = ip->addrs[NDIRECT + 1]) == 0)
404       ip->addrs[NDIRECT + 1] = addr = balloc(ip->dev);
405
406     bp = bread(ip->dev, addr);
407     indirect = (uint*)bp->data;
408     indirect_idx = bn / NINDIRECT;
409
410     if((addr = indirect[indirect_idx]) == 0){
411       addr = indirect[indirect_idx] = balloc(ip->dev);
412       log_write(bp);
413     }
414
415     bp2 = bread(ip->dev, addr);
416     double_indirect = (uint*)bp2->data;
417     double_indirect_idx = bn % NINDIRECT;
418
419     if((addr = double_indirect[double_indirect_idx]) == 0){
420       addr = double_indirect[double_indirect_idx] = balloc(ip->dev);
421       log_write(bp2);
422     }
423
424     brelse(bp2);
425     brelse(bp);
426     return addr;
427   }
428
429   panic("bmap: out of range");
430 }
```

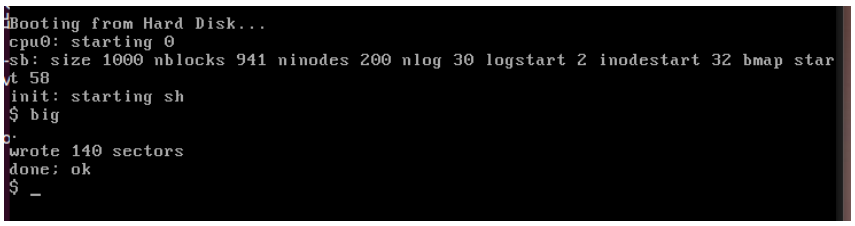
```
#define NDIRECT 11
#define NINDIRECT (BSIZE / sizeof(uint))
#define MAXFILE (NDIRECT + NINDIRECT + NINDIRECT * NINDIRECT)

// On-disk inode structure
struct dinode {
  short type;           // File type
  short major;          // Major device number (T_DEV only)
  short minor;          // Minor device number (T_DEV only)
  short nlink;          // Number of links to inode in file system
  uint size;            // Size of file (bytes)
  uint addrs[NDIRECT+2];   // Data block addresses
};
```

三、编译、运行、测试说明（简单说明如何编译、运行、测试你提交的代码。如果程序由多个源程序构成，建议编写 Makefile，或者给出编译脚本。）

修改前后均需要在输入 make qemu 指令后，运行 big 命令

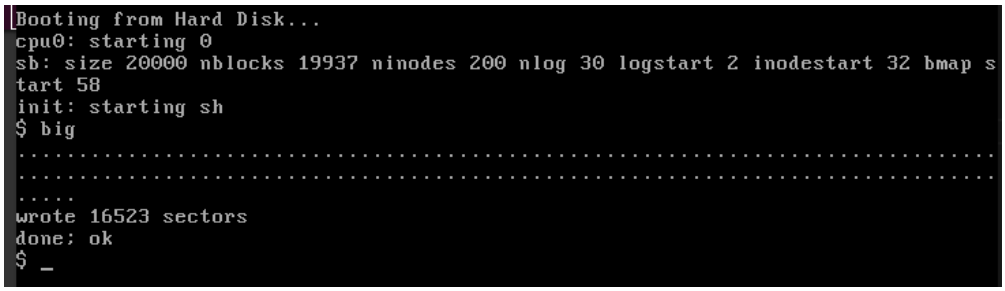
四、实验结果与结论分析（经调试正确的程序的运行结果截图，包括输入数据、输出结果、结论）

1、修改 inode 数据结构定义之前运行结果：



xv6 索引节点包含 12 个直接索引和一个一级间接索引。sector 数量=(12+(512/4))=140 个，因此能支持的最大文件大小为 140*512=71680 字节。

2、修改 inode 数据结构定义之后的运行结果：



xv6 索引节点包含 11 个直接索引、一个一级间接索引和一个二级间接索引。sector 数量=(11+(512/4)+(512/4)*(512/4))=16523 个，因此能支持的最大文件大小为 16523*512=8459776 字节。

五、实验中遇到的问题及解决方法

完成日期：2020.12.29