Q1.

1. 文件以可读可写的模式打开，

sprintf(buf, "%3d\n", i);

n = write(fd, buf, strlen(buf)); printf("len=%d\n", strlen(buf));

在每个循环内部首先以4字节宽度将i读入buf（sprintf），然后将该值写入tmpfile，上述过程重复1000次。

于是总共使用的内存没有超过一块，于是tmpfile的大小为4KB

1. 没有文件缓存的情况下，需要先访问当前目录所对应的dentry所在的数据块，创建属于tmpfile的dentry，写一次磁盘，大小为一个数据块；为其分配一个磁盘块，分配inode节点，写一次磁盘，大小为一个数据块；修改inodemap和blkmap，写两次，每次写一个扇区。

随后进行1000次写操作，每次都要写4KB，虽然每次只修改4B的内容，但是写穿透的粒度为

1. 有文件缓存的情况下，总共写1次磁盘，大小为4KB。

Q2.

创建一个文件，首先读取根目录，访问/home的inode节点，读取指针指向的数据块并查找OS18，查找到0S18后访问OS18的inode，读取指针指向的数据块，然后在数据块中创建fs03.pdf的dentry，此时需要写该数据块对应的缓存；然后创建fs03.pdf的inode，此时需要写inode数据块对应的缓存；创建pdf文件要使用的数据块；修改inodemap和blkmap对应的数据块对应的缓存块。至于如何替换缓存块取决于具体的算法。

宕机时，当宕机发生在所有这些缓存块回写之前的话，实际上是什么也没有修改。

如果inodemap写回成功而blkmap未被修改，则下一次使用时pdf文件使用的块可能会被分配给别的文件。

如果inodemap写回失败，那么pdf文件的inode可能会被下一个生成的文件所替换。

如果inodemap写回成功，blkmap写回成功，但是OS18指针指向的数据块未写回成功，那么pdf文件的dentry不能通过路径找到。

如果只有inode数据块写回失败，那么pdf文件的inode节点生成失败，该空间被浪费。

Q3.

1. 数据日志的顺序有：写日志块（TxB，inode日志，bitmap日志，数据块日志），提交日志（TxE），Checkpoint（修改），清除日志。

宕机如果发生在提交日志之后，那么宕机恢复之后A文件的内容都是修改过0块和1块的内容。

宕机如果发生在提交日志之前，那么A文件的内容都是修改之前的内容。

1. 采用元数据日志的顺序是：写数据块，写日志（TxB，inode日志，bitmap日志），提交日志，checkpoint，清除日志。

宕机如果发生在提交日志之后，那么宕机恢复之后A文件的内容都是修改过0块和1块的内容。

宕机如果发生在写数据块之前，那么A文件的内容都是修改之前的内容。

宕机如果发生在写数据块中，那么A文件中的0块和1块从宕机操作位置分开，之前为修改过的部分，之后为未修改过的部分。

宕机如果前发生在写数据块与写日志之间，那么A文件的完整性不会受到影响。

Q4.

1. 磁盘地址为4B，500万个inode占用imap空间5000000\*4B = 20000000B

一个块4KB，共使用20000000B/4096B（向上取整） = 4883个块。

1. CR记录每个imap块的磁盘地址，4883个块占用CR空间4883\*4B = 19532B

共使用CR 19532B/4096B （向上取整）= 5个块。

1. 一个imap块可存放1024个inode的地址，故ino=654321在654321/1024（向上取整）=第639个imap块中的第1009个项中。而一个CR块可存放1024个imap块地址，所以第639个imap块的首地址在第一个CR块中的第639项中。

Q5.

1. FFS采用三级间址。Foo大小20MB，占用5个segment，或者说占用5120个块。一个inode节点中的一级指针有1024个，一个一级指针下的二级指针有1024个，一个二级指针下的三级指针有1024个，所以20MB大小的文件的各个块地址都可以在inode下的第一个一级指针的第一个二级指针下的5120个三级指针中找到。Inode的地址可由imap找到，imap的地址可由CR找到。
2. LFS的修改结构是不覆写顺序修改。读取foo的inode，先读取一级指针块，二级指针块，三级指针块，找到2560块所在的地址Ai，将该块的内容读到内存中，然后修改该内容，之后顺序写到磁盘中的末尾处，然后写修改后的inode所在块，然后写修改后的imap所在块，然后修改CR块。

写在磁盘上的顺序：位于开头的CR块，数据块，inode块，imap块

需要7次In，4次Out

1. FFS需要写foo的第2560块，inode块，一共需要写两个块，因为inodemap和blkmap都不需要写。读取foo的inode，然后先读取一级指针块，二级指针块，三级指针块，找到2560块所在的地址，将该块的内容读到内存中，然后修改该内容，然后写回磁盘，修改inode，写回磁盘。

需要5次In，2次Out

1. 元数据修改日志写操作顺序为写数据块，写日志（TxB，inode日志），提交日志，checkpoint，清除日志。注意，inodemap、blkmap不需要写。

故需要读三个指针块，一个数据块，一个inode块。5次In

需要写三个日志块，inode块，清除日志块。5次Out

PS：老师，该次作业很多对文件的访问要求都不明确，所以对于同一个题目，默认有哪些已知条件的不同可能导致结果的不同，比如说读取块的次数或者I/O的次数，很有可能与规定的答案不同。