

Aayush Kumar
CO21BTECH11002
File System Assignment

Part 1:

```
$ echo > a1  
log_write 34  
log_write 34  
log_write 59
```

- In the first disk write, `ialloc` is called. It allocates an inode on the device and marks it as allocated.
- In the second disk write, `iupdate` is called. It copies the in-memory inode to disk.
- Finally, `writei` is called where the function writes data into the inode.

```
$ echo x > a1  
log_write 58  
log_write 644  
log_write 644  
log_write 34  
log_write 644  
log_write 34
```

- In the first disk write, `balloc` is called and it allocates a disk block.
- Next `bzero` is called and it zeros the allocated block.
- The next calls are of `writei` and `iupdate` which writes the data and copies it into the file.
- Some calls are repeated because of how `xv6` writes data into files.

```
$ echo x > a1  
log_write 644  
log_write 644
```

- On repeating the same call, we get only two calls to `writei`.
- This is since we have already allocated memory to file and its contents and we only need to update it.

```
$ echo xxx > a1
log_write 644
log_write 644
log_write 644
log_write 34
log_write 644
log_write 34
```

- This contains calls to only writei and iupdate.
- First previous data is removed from the file and the inode is updated.
- The new data is written and copied into the file and the inode is updated with the new information.

```
$ rm a1
log_write 59
log_write 34
log_write 58
log_write 34
log_write 34
```

- This contains calls to writei, iupdate and bfree.
- The file data is erased, inode is updated and the block allocated to a1 is freed.

```
$ echo y > a2
log_write 34
log_write 34
log_write 59
log_write 58
log_write 644
log_write 644
log_write 34
log_write 644
log_write 34
```

- Here making the file and updating its content is happening simultaneously.
- First ialloc is called, it allocates an inode corresponding to the file.
- Next iupdate updates the inode.
- Next writei is called.
- Next balloc and bzero are called. It allocates and zeros a block of memory for storing the file content.
- Next writei and iupdate are each called twice. They write the file content and update the inode.
- Block 34 is allocated to a2.

Part 2:

```
$ echo x > a1  
log_write 34  
log_write 34  
log_write 59  
log_write 58  
log_write 644  
log_write 644  
log_write 34  
log_write 644  
log_write 34
```

```
$ echo y > a2  
log_write 34  
log_write 34  
log_write 59  
log_write 58  
log_write 645  
log_write 645  
log_write 34  
log_write 645  
log_write 34
```

```
$ echo z > a3  
log_write 34  
log_write 34  
log_write 59  
log_write 58  
log_write 646  
log_write 646  
log_write 34  
log_write 646  
log_write 34
```

- In this case, all the blocks allocated for creating different files are the same except for the block storing file contents.
- This is because as the files are created, the same memory blocks are used for storing the temporary file information.
- Since the files are different, the block storing file information is different in each case.

```
$ rm a1 a2 a3  
log_write 59  
log_write 34  
log_write 58  
log_write 34  
log_write 34  
log_write 59  
log_write 34  
log_write 58  
log_write 34  
log_write 34  
log_write 59  
log_write 34  
log_write 58  
log_write 34  
log_write 34
```

- Here for each deletion the same blocks are used to store the temporary information about the files.
- Hence the same sequence of blocks are repeated 3 times.