

# **Part 1**

1. Output for echo > a1:

```
log_write 34  
log_write 34  
log_write 59
```

2. Output for echo x > a1:

```
log_write 58  
log_write 644  
log_write 644  
log_write 34  
log_write 644  
log_write 34
```

3. Output for echo xxx > a1:

```
log_write 644  
log_write 644  
log_write 644  
log_write 34  
log_write 644  
log_write 34
```

4. Output for rm a1:

```
log_write 59  
log_write 34  
log_write 58  
log_write 34  
log_write 34
```

5. Output for 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
```

**Explanation:**

1. Our root directory from where we are performing all these operations is on block 34. Therefore we are seeing number 34 come up again and again. The operations are happening in the following order:
  - Allocate a new inode for the new file.
  - Update the inode of the directory.
  - Link the directory and file since the file is referencing the directory
2. The following operations are happening in order:
  - Allocate a block of memory for the file. Block 58 tracks which blocks have been allocated on disk.
  - Zero out the allocated block. Block 644 is zeroed out. File data is here.
  - Write to block 644.
  - Update inode of the file for this update.
  - Write end of file to block 644.
  - Update inode of the file for this update.
3. The following operations are happening in order:
  - Write x to block 644
  - Write x to block 644
  - Write x to block 644
  - Update inode of file
  - Write end of file to block 644
  - Update inode of file
4. The following operations are happening in order:
  - Unlink the file link with directory
  - Update the inode of the directory
  - Free the block of memory allocated to the file
  - Truncate the inode of the file.
  - Update the inode of the directory.
5. The following operations are happening in order:
  - Allocate a new inode
  - Update directory inode
  - Link directory and file
  - Allocate a block of memory
  - Zero out the allocated block of memory
  - Write x to the block of memory
  - Update inode of file
  - Write end of file to block of memory
  - Update inode of file

## **Part 2**

Output for 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
```

Output for 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
```

Output for echo y > a2:

```
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
```

Output for 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

**Explanation:**

The explanation for the first three commands is the same as the fifth command in part 1. The following operations are happening in order:

- Allocate a new inode
- Update directory inode
- Link directory and file
- Allocate a block of memory
- Zero out the allocated block of memory
- Write x to the block of memory
- Update inode of file
- Write end of file to block of memory
- Update inode of file

The only difference we see is that block 644 gets replaced by block 645 and 646. This is because each file is getting allocated a new block of memory. Rest outputs are the same because many times, we are printing the block number of the directory, which is block 34. Also, block 59 contains inodes and block 58 contains information about allocated blocks on disk. Thus blocks 34, 58, 59 are the same for all three files. Note that block 34 is the same because all three files are being created in the same directory.

For the last command, the explanation is the same as the fourth command in part 1. The following operations are happening in order:

- Unlink the file link with directory
- Update the inode of the directory
- Free the block of memory allocated to the file
- Truncate the inode of the file.
- Update the inode of the directory.

First five outputs are for file a1, next five for a2 and last five for a3. Note that the outputs are the same for all three files. This is because again, 34, 58, 59 blocks are not changing for the files.