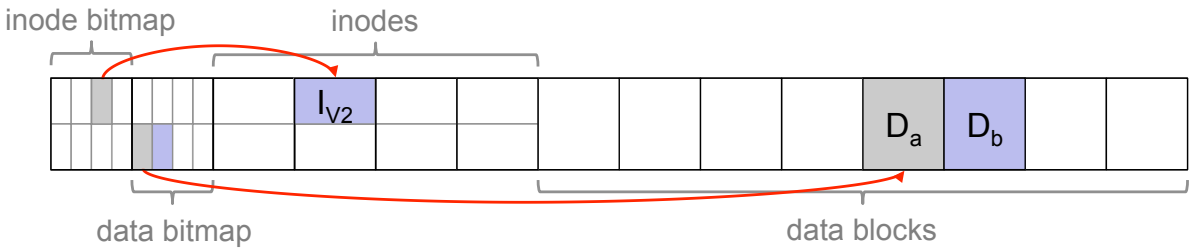
# Journaled File Systems

- What if systems crash?
- *Post-mortem* recovery: File system checkers
- Pro-active "recovery": Journaling
- Data vs. Meta-Data Journaling
- Journaling and Block Re-use

# What can happen when System crashes?

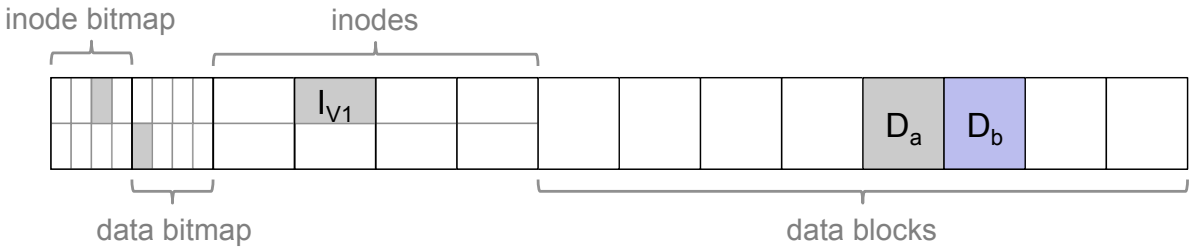



Append Block to File:

1. Write data block
2. Update data bitmap
3. Update i-node

Examples from:
Operating Systems: Three Easy Pieces
Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau

# What can happen when System crashes?



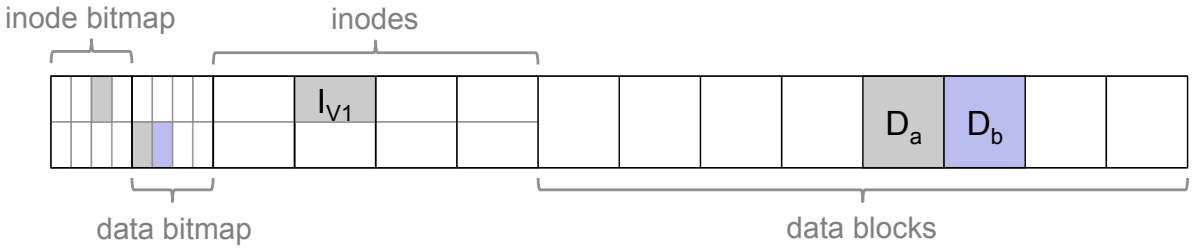

Append Block to File:

1. Write data block
2. Update data bitr
3. Update i-node


Effect of Crash:

1. Data block has been written.
2. Change not recorded in inode or bitmap
3. File system remains consistent.
4. User looses update.

# What can happen when System crashes?



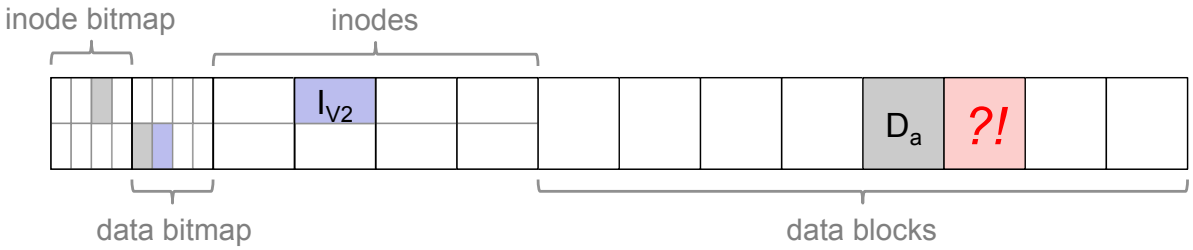

Append Block to File:

1. Write data block
2. Update data bitr
3. Update i-node


Effect of Crash:

1. Data block has been written.
2. Change recorded in bitmap.
3. Change not recorded in inode.
4. System leaks Block #5.

# What can happen when System crashes?



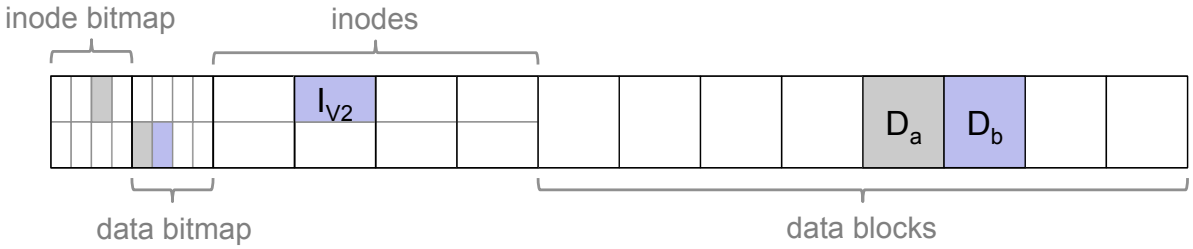

Append Block to File:

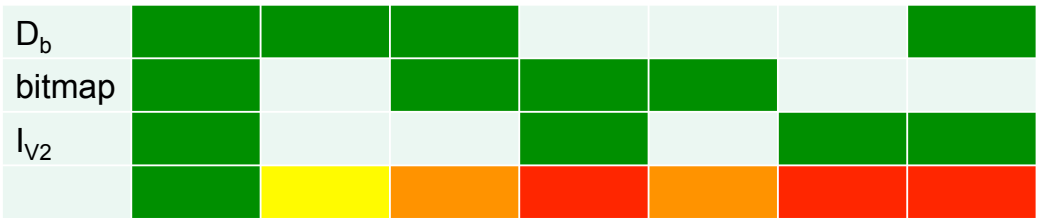
1. Write data block
2. Update data bitr
3. Update i-node


Effect of Crash:

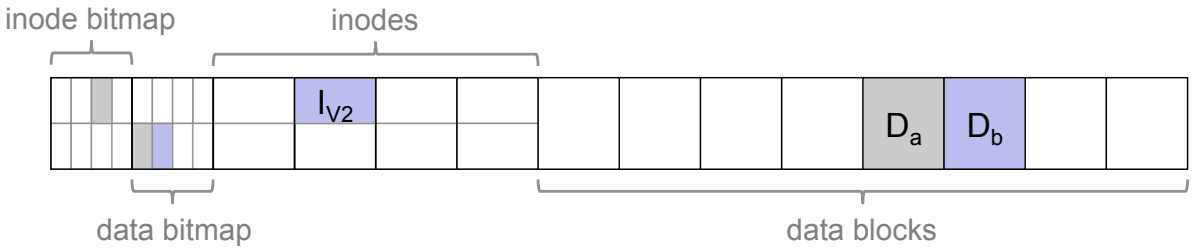
1. Change recorded in bitmap.
2. Change recorded in inode.
3. Block #5 not written.
4. File System inconsistent!

# What can happen when System crashes?



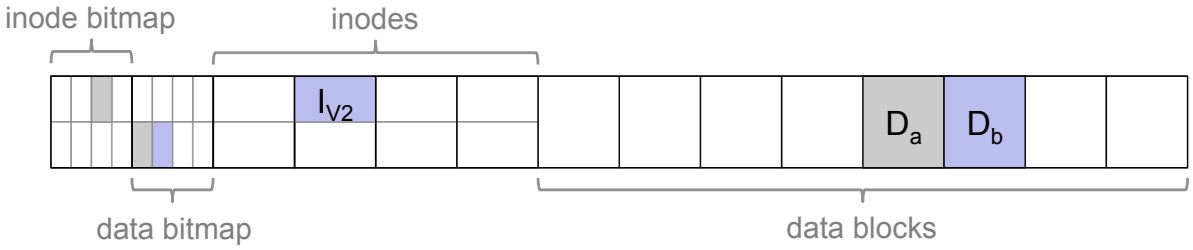



# What can happen when System crashes?




Can we recover from these crashes?


# *Post Mortem* Recovery: File System Checkers




***Post-Mortem* File System Checker:**

1. Check if superblock is ok.
2. Scan i-nodes -> new data bitmap
3. New inode bitmap
4. Check state of inodes
5. Check link counters in inodes
6. Check duplicate pointer to blocks
7. Check for bad block pointers
8. Check directories

# *Post Mortem* Recovery: Pros can Cons

***Post-Mortem* File System Checker:**

1. Check of superblock is ok.
2. Scan i-nodes -> new data bitmap
3. New inode bitmap
4. Check state of inodes
5. Check link counters in inodes
6. Check duplicate pointer to blocks
7. Check for bad block pointers
8. Check directories

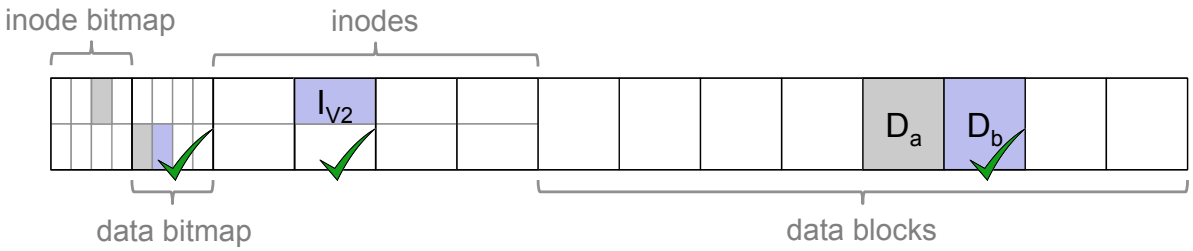
**Pros:**

- No overhead during normal disk operation

**Cons:**

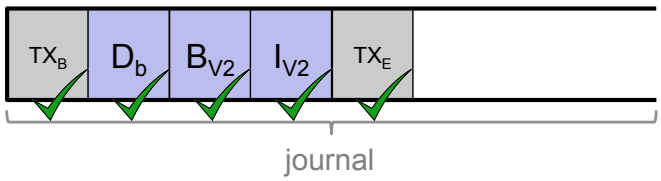
- Requires very detailed information about file system
- Cannot correct all types of errors
- Expensive!

# Pro-Active "Recovery": Journaling

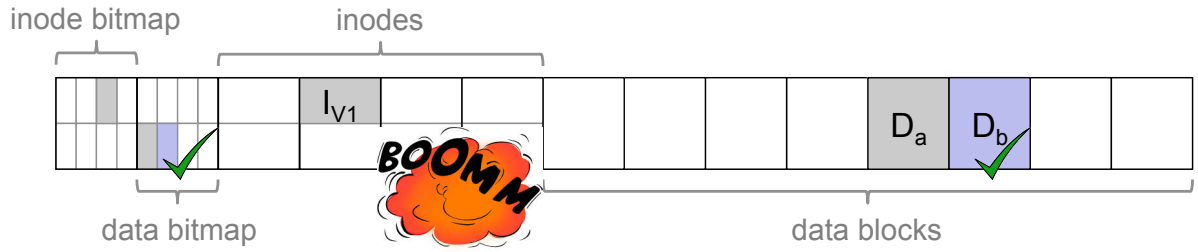



Append Block to File:

1. Write data block
2. Update data bitmap
3. Update i-node

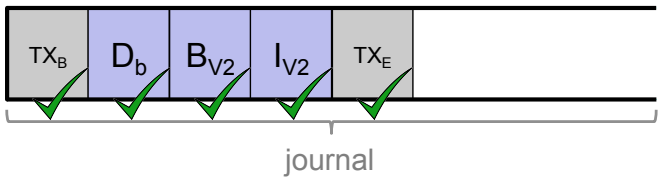


# Pro-Active "Recovery": Journaling

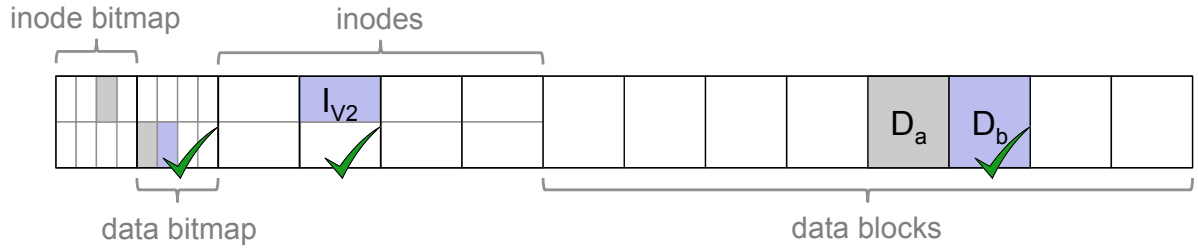



Append Block to File:

1. Write data block
2. Update data bitmap
3. Update i-node

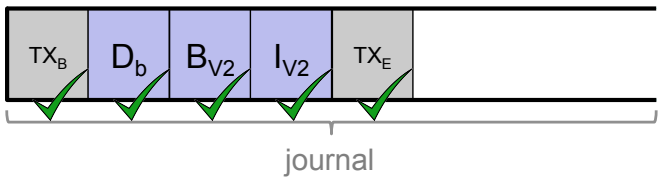



# Pro-Active "Recovery": Journaling

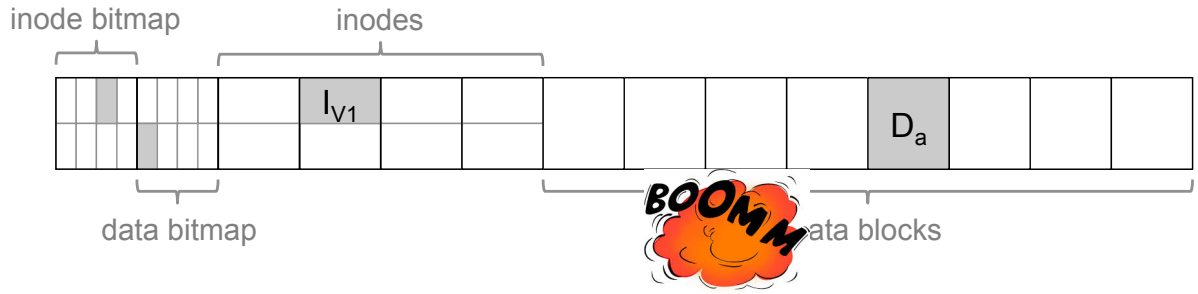



Append Block to File:

1. Write data block
2. Update data bitmap
3. Update i-node

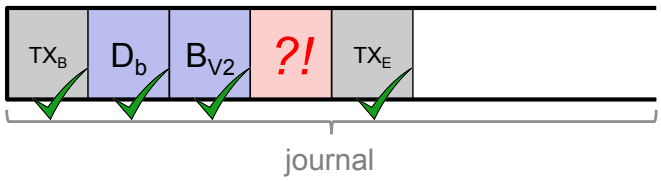


# Pro-Active "Recovery": Journaling

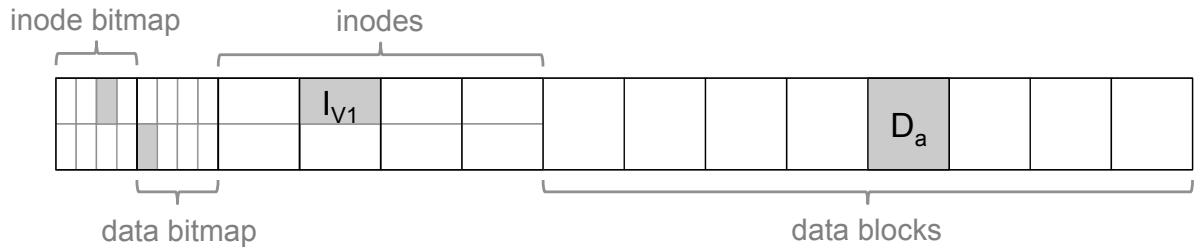



Append Block to File:

1. Write data block
2. Update data bitmap
3. Update i-node

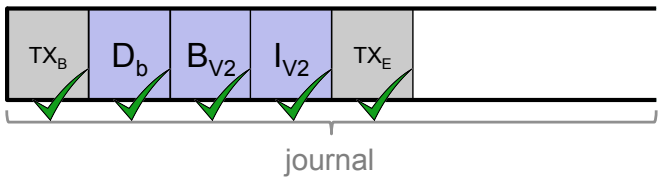



# Pro-Active "Recovery": Journaling

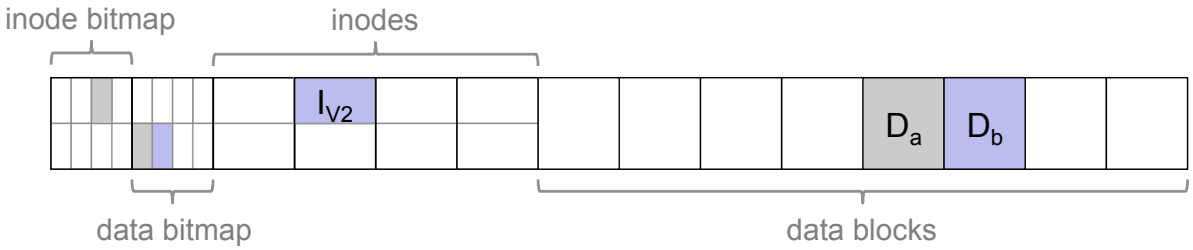



Append Block to File:

1. Write data block
2. Update data bitmap
3. Update i-node



# Meta-Data Journaling

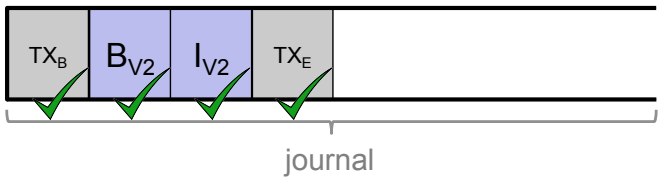



**Observation:**

Write operations are slow!

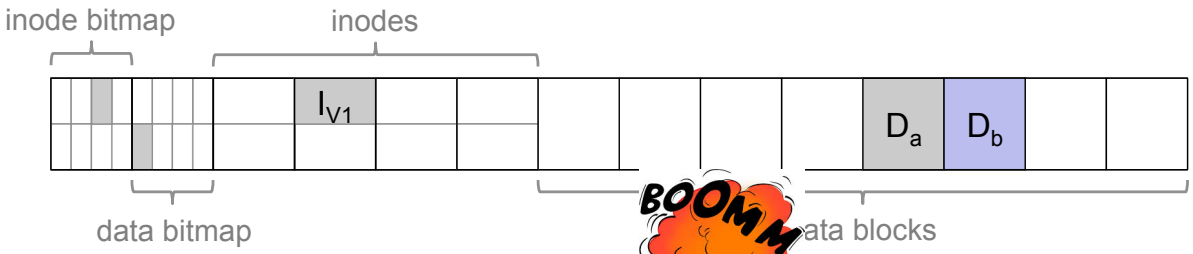
For each write, we write to journal as well.

This doubles the write operations.




**Solution:** Journal meta data only!

# Meta-Data Journaling

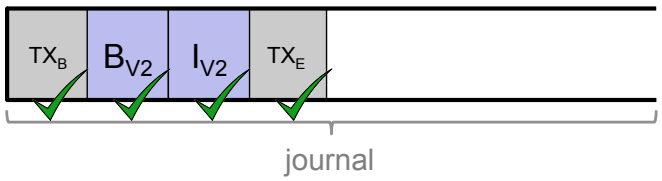



**Observation:**

Write operations are slow!

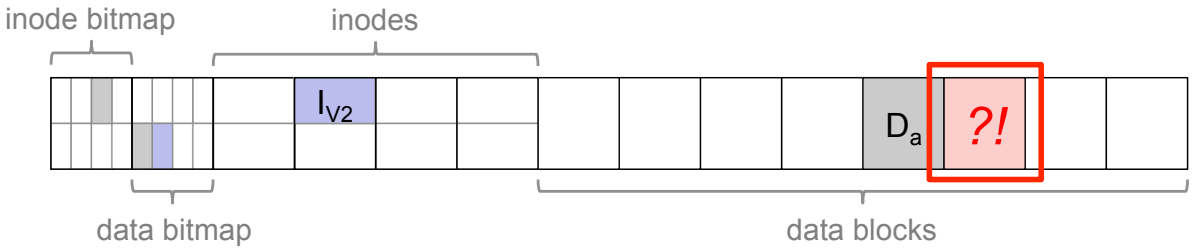
For each write, we write to journal as well.

This doubles the write operations.




**Solution:** Journal meta data only!

# Meta-Data Journaling

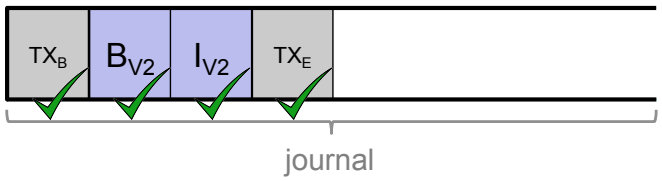



**Observation:**

Write operations are slow!

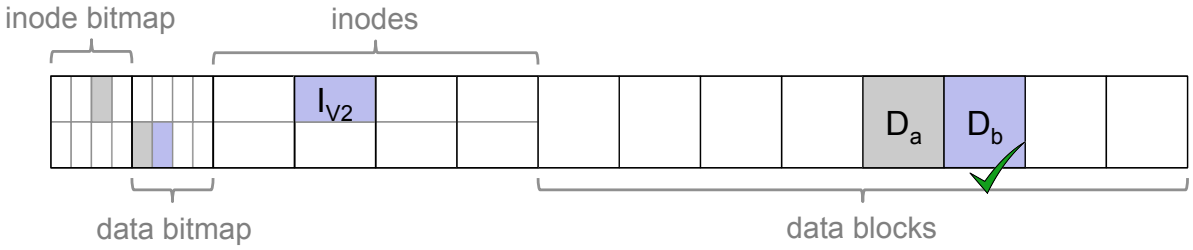
For each write, we write to journal as well.

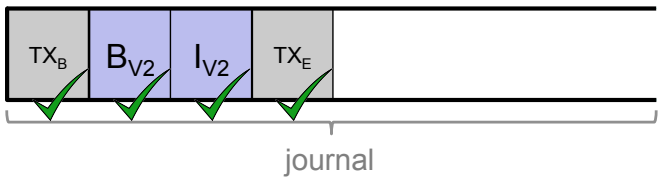
This doubles the write operations.




**Solution:** Journal meta data only!

# Meta-Data Journaling

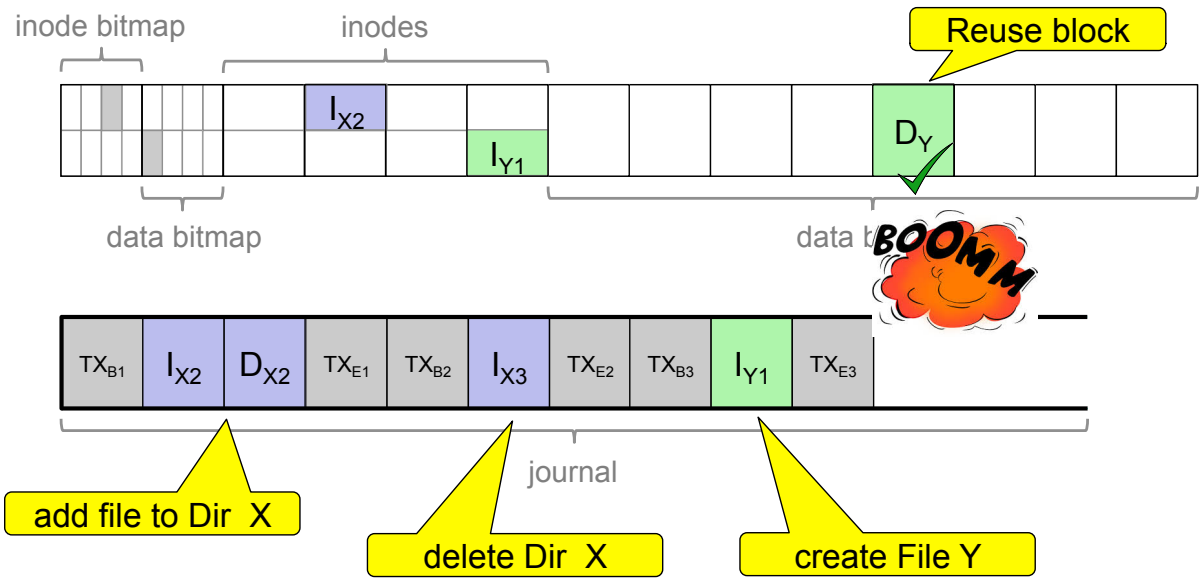


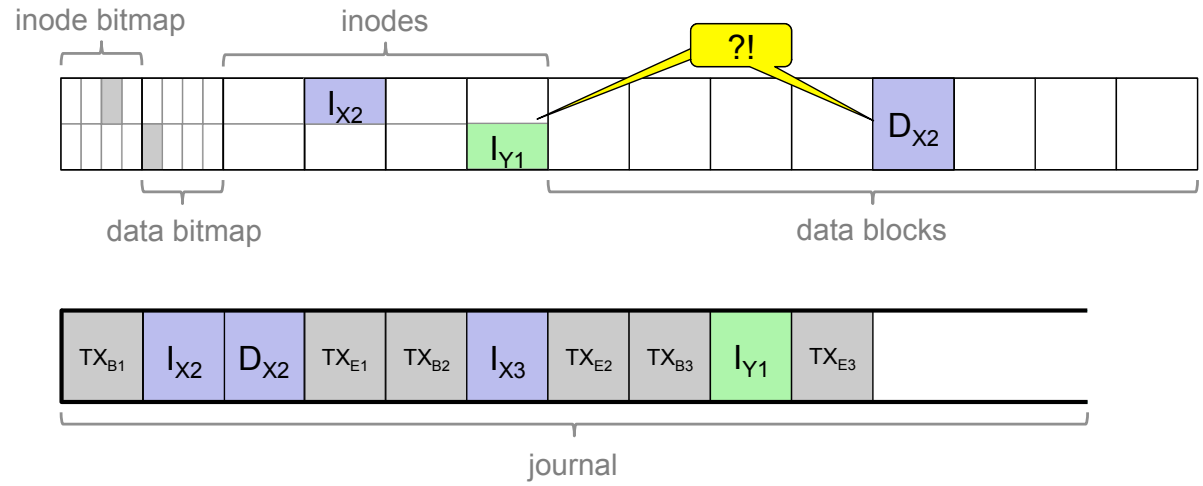

**Observation:**

Ordering of operations is important!

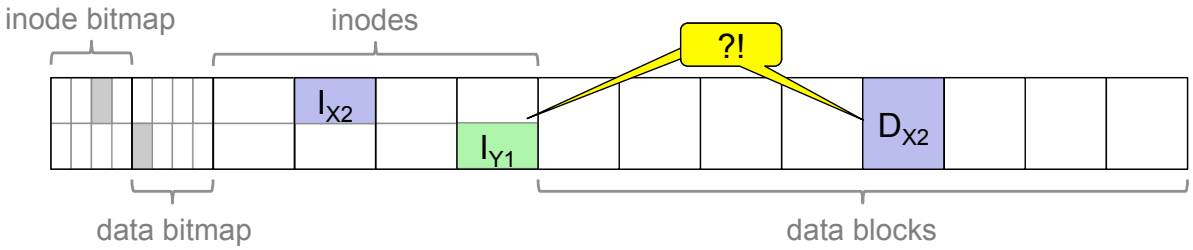
# Journaling: Dealing with Block Reuse




# Journaling: Dealing with Block Reuse



# Journaling: Dealing with Block Reuse




**Solutions:**

- Do not re-use blocks until after "delete" purged from journal.
- Linux ext3: special "revoke" records, which prevent data blocks to be written to disk.

# Journaled File Systems

- What if systems crash?
- *Post-mortem* recovery: File system checkers
- Pro-active "recovery": Journaling
- Data vs. Meta-Data Journaling
- Journaling and Block Re-use