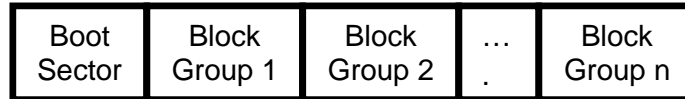


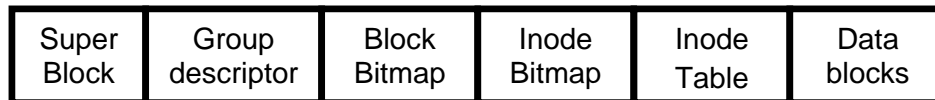
## Ext2 File System in a Nutshell

vineet Agarwal ([checkout.vineet@gmail.com](mailto:checkout.vineet@gmail.com))

### Physical structure of Ext2 File System:



### Structure of Block group in an Ext2 File System:



<-This is same for all block groups -> <-----This is specific to each group----->

The file system is created from a sequential collection of blocks. These blocks can be 1k, 2k or 4k in size. These blocks are divided up into groups. The starting point for the file system is the superblock and is always located at an offset of 1024 bytes from the start of the file system.

The information in the superblock is used to access the rest of the data on the disk.

- **Number of block groups =  $\text{block\_count} / \text{blocks\_per\_group}$**

The first block on the disk is block 1

- **Block address of each block group =  $((\text{group\_number} - 1) * \text{blocks\_per\_group})$**

Where group\_number is the number of the block group you want to access.

The Group Descriptors contains information on the block groups. This data is covers all the groups and is stored in all the groups for redundancy.

- **Size of descriptors in blocks =  $(\text{sizeof}(\text{struct ext2\_group\_desc}) * \text{number\_of\_groups}) / \text{block size}$**

The block bitmap is a bitmap indicating which blocks in the group have been allocated. If the bit is set then the block is allocated.

- **size of the block\_bitmap =  $(\text{blocks\_per\_group} / 8) / \text{block\_size}$**
- **Group of particular data block =  $((\text{block\_number} - 1) / \text{s\_blocks\_per\_group}) + 1$**
- **The block in that group (offset in blocks) =  $\text{block\_number} \% \text{s\_blocks\_per\_group}$**

The inode bitmap indicates which inodes are allocated. If the bit is set then the inode is allocated.

- **Size of the inode bitmap** = (inodes\_per\_group / 8) / block\_size
- **Group of particular inode** = ((inode\_number - 1) / s\_inodes\_per\_group) + 1
- **The inode in that group (offset in blocks)** = inode\_number % s\_inodes\_per\_group

Inode table

**No . of blocks in inode table** = inode\_per\_group \* 128 / blocksize

## **20 MB Partition meta-data layout**

offset	# of blocks	description
0	1	boot record
-- block group 0 --		
(1024 bytes)	1	<u>superblock</u>
2	1	<u>group descriptors</u>
3	1	<u>block bitmap</u>
4	1	<u>inode bitmap</u>
5	214	<u>inode table</u>
219	7974	<u>data blocks</u>
-- block group 1 --		
8193	1	<u>superblock backup</u>
8194	1	<u>group descriptors backup</u>
8195	1	<u>block bitmap</u>
8196	1	<u>inode bitmap</u>
8197	214	<u>inode table</u>
8408	7974	<u>data blocks</u>
-- block group 2 --		
16385	1	<u>block bitmap</u>
16386	1	<u>inode bitmap</u>
16387	214	<u>inode table</u>
16601	3879	<u>data blocks</u>

## **Explanation:**

Do mke2fs on 20 MB partition

Use superblock information

To get the information of the no. of Block group = s\_block\_count / s\_blocks\_per\_group

$$bg = 20480 / 8192 = 2.5$$

So. No of block group = 2.5

Super block start from the offset 1024.

$$\text{Group desc} = (32 * 2.5) / 1024 = .078$$

it would take just one block at minimum

$$\text{No. of blocks for block bitmap} = (8192 / 8) / 1024 = 1$$

$$\text{No. of blocks for inode bitmap} = (1712 / 8) / 1024 = 0.204$$

So minimum it takes 1 block

$$\text{No. of blocks for inode table} = (1712 * 128) / 1024$$

Where Inodes per group = 1712 and Inode size = 128

No of data blocks = remaining blocks i.e.  $\text{blocks\_per\_group} - \text{Total allocated}$

Similarly it is done for all block groups apart from last block group as it might not be complete.