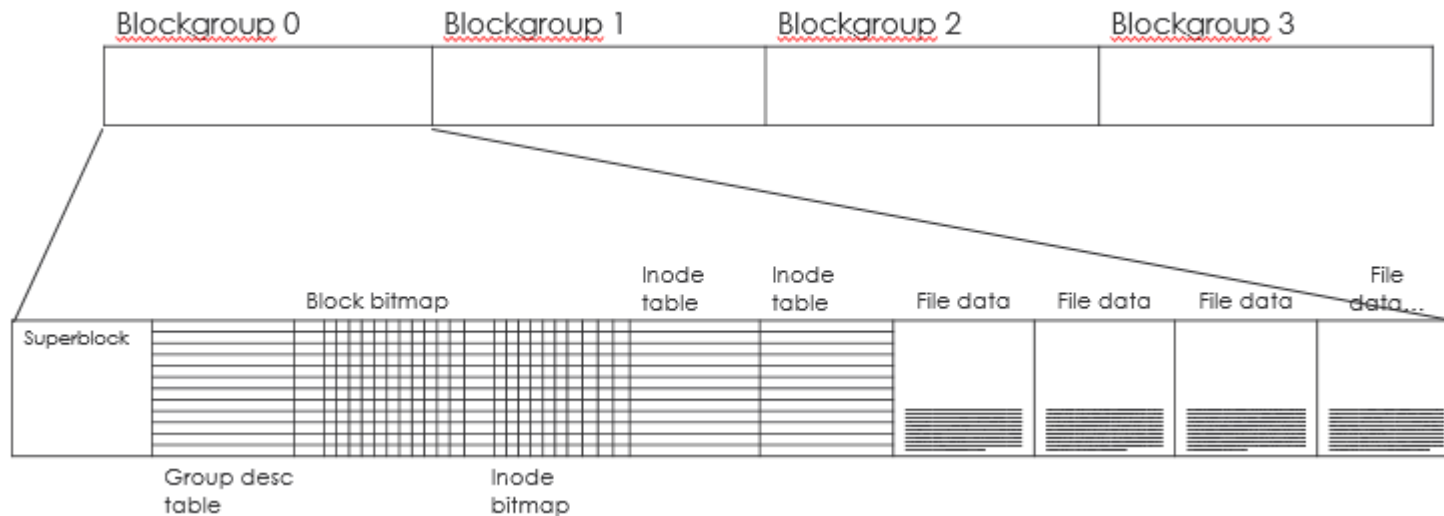


Ext 2/3

Volume layout

(blockgroup count is variable)



Superblock (0x400 bytes long. In theory can be at offset 0x400, 0x800, 0xB00 or 0xD00. Most likely 0x400)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Num inodes			Num blocks				Reserved blocks				Num unallocated blocks				
10	Num unallocated inodes			Block group 0				Block size ¹				Fragment size				
20	Blocks per block group			Fragments per block group				Inodes per block group				Last mount time				
30	Last written time			Mount count	Max mount count		53	EF	Filesystem state		Error handling	Minor version				
40	Last consistency check time			Consistency check frequency				Creator OS				Major Version				
50	UID reserve permissions	GID reserve permissions		First non-reserved inode				Inode size		Block group of superblock		Optional feature flags				
60	Required feature flags			Read only feature flags				File system ID								
70	File system ID (cont)								Volume name							
80	Volume name (cont)								Last mount point							
90	Last mount point (cont)															
A0	Last mount point (cont)															

B0	Last mount point (cont)				
C0	Last mount point (cont)	Algorithm usage bitmap	*2	*3	unused
D0	Journal ID				
E0	Journal inode	Journal device	Start of orphan inode list	Hash seed	
F0	Hash seed			*4	
100	Default mount options	First meta BG			
...	unused				
3F0	unused				

1 – 0x00 = Bitwise left shift value applied to 1024 to give block size. 0 = 1024, 1 = 2048, 2 = 4096

***2 -Number of blocks preallocated for files**

***3 – Number of block preallocated for directories**

***4 - Hash version**

Note that most of these values are considered “non-essential” and may not be assigned by any given operating system. Coloured values are required.

Group Descriptor Table Entry (located at first block after superblock)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	First block of block bitmap			First block of inode bitmap			First block of inode table			Num unallocated blocks			Num unallocated inodes			
10	Num directories		unused													

Inode

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	File mode	Lower User ID		Lower size				Access Time				Change Time				
10	Modification Time			Deletion Time				Lower Group ID		Link count		Sector count				
20	Flags			Unused				Direct block pointer			Direct block pointer					
30	Direct block pointer			Direct block pointer				Direct block pointer			Direct block pointer					
40	Direct block pointer			Direct block pointer				Direct block pointer			Direct block pointer					
50	Direct block pointer			Direct block pointer				Singly indirect block pointer			Doubly indirect block pointer					
60	Triply indirect block pointer			Generation number				Extended attribute block			Upper file size					
70	Block address of fragment			*1	*2	Reserved		High User ID		High Group ID		reserved				

More on Inode data

Value	Size	offset	Comments
File mode	2	0x00	The file mode sets file permissions and features of the file. See next table
Lower user ID	2	0x02	Original versions of EXT2 used 16 bits for user ID, more recent versions support 32 bits.
Lower size	4	0x04	32bits for size, another 32 bits for size at 0x6C
Access Time	4	0x08	Last time the file contents were accessed.
Change time	4	0x0C	Last time the inode entry changed
Modification time	4	0x10	Last time the file contents changed
Deletion time	4	0x14	Time the file was deleted
Lower group ID	2	0x18	16 bits for lower half of group ID
Link count	2	0x1A	Count of links (e.g. directory entries) pointing to the file. Generally 1, but can be higher. Symbolic links do not affect count.
inode block count	4	0x1C	Number of 512 blocks used by the inode
Flags	4	0x20	Attributes that are set for the file. Behavior is dependent upon supported attributes of filesystem & OS.

Value	Size	offset	Comments
Direct block pointers	12*4	0x28	pointers to a block containing file data
Indirect block pointer	4	0x58	pointer to a block containing pointers to file data
Doubly indirect block pointer	4	0x5C	pointer to a block containing pointers to blocks containing pointers to file data
Triply indirect block pointer	4	0x60	pointer to a block contains pointers to blocks containing pointers to blocks containing pointers to file data
Generation number	4	0x64	Number indicating file version (not widely supported)
Extended attribute block/file ACL	4	0x68	Varying levels of support, enables storing of name/value pairs as provided by applications.
Upper file size / dir ACL	4	0x6C	upper 32 bits of file size, or extended directory access control list.
Block address of fragment	4	0x70	not widely implemented, possibly unused.
Fragment index in block	1	0x74	
Fragment size	1	0x75	
High user ID	2	0x78	high 32bits of UID
High group ID	2	0x7A	high 32bits of GID

Inode file mode

Meaning	Binary	hex
Other execute	0000000000000001	0x0001
Other write	0000000000000010	0x0002
Other read	0000000000000100	0x0004
Group execute	0000000000001000	0x0008
Group write	0000000000010000	0x0010
Group read	0000000000100000	0x0020
Owner execute	0000000001000000	0x0040
Owner write	0000000010000000	0x0080
Owner Read	0000000100000000	0x0100
Sticky bit	0000001000000000	0x0200
Set process GID	0000010000000000	0x0400
Set process UID	0000100000000000	0x0800
FIFO	0001000000000000	0x1000
Character device	0010000000000000	0x2000
Directory	0100000000000000	0x4000
Block device	0110000000000000	0x6000
Regular file	1000000000000000	0x8000
Symbolic link	1010000000000000	0xA000
Socket	1100000000000000	0xC000

Directory entry

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Inode value			Entry len	Name len	File type	File name... (variable length)									

Directory entry file types

Meaning	Value
Unknown	0
Regular file	1
Directory	2
Character device	3
Block device	4
FIFO	5
Unix socket	6
Symbolic link	7