

Figures for Chapter 10

10 Regular file systems

- 10.1 An entry in the linked list of file systems
- 10.2 Layout of a disk
- 10.3 Contiguous allocation
- 10.4 Linked allocation
- 10.5 Example of a file map
- 10.6 Layout of a partition with Ext2
- 10.7 A block group on disk
- 10.8 An Ext2 super block
- 10.9 Structure of a block group descriptor
- 10.10 Structure of an Ext2 disk inode
- 10.11 Allocation information in an inode
- 10.12 Single indirect block
- 10.13 Double indirect block
- 10.14 Fragments of two files in a block
- 10.15 An Ext2 directory entry
- 10.16 File systems as they exist on disk
- 10.17 Directory structure after attaching file system B
- 10.18 Data structure representing a mounted file system
- 10.19 Operations on a file system
- 10.20 Superblock for an Ext2 file system
- 10.21 Data structures involved in mounting a file system
- 10.22 Ext2 file system inode data in memory


```
struct file_system_type{
    const char          *name;
    struct super_block  *(*read_super)();
    struct file_system_type *next;
};
```

Figure 10.1: An entry in the linked list of file systems

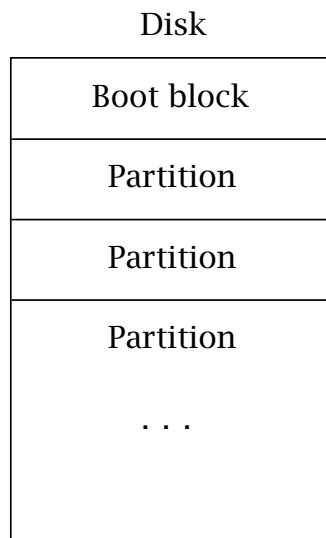


Figure 10.2: Layout of a disk

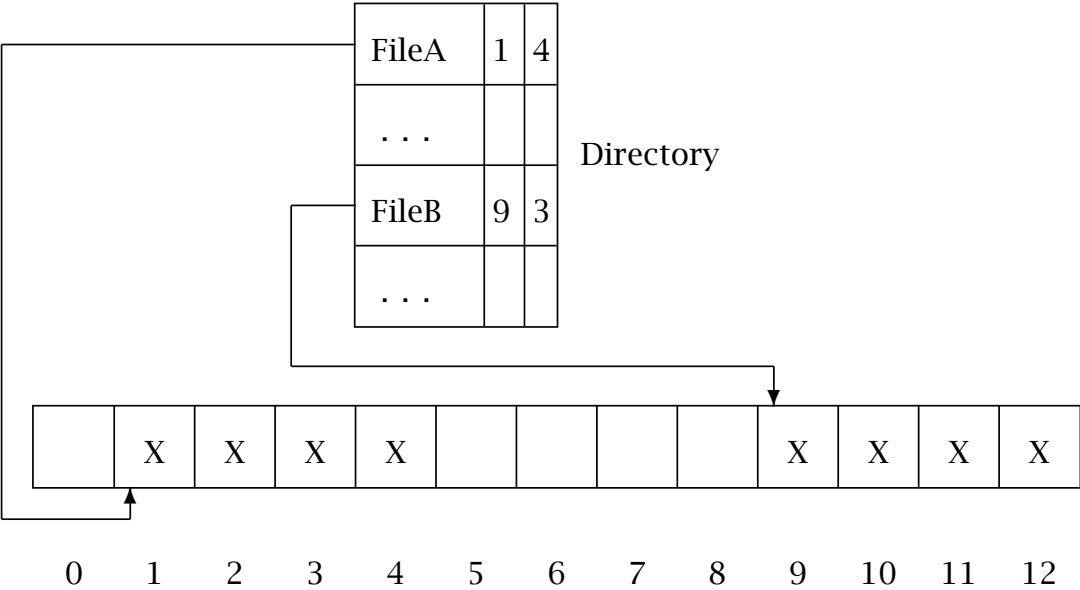


Figure 10.3: Contiguous allocation

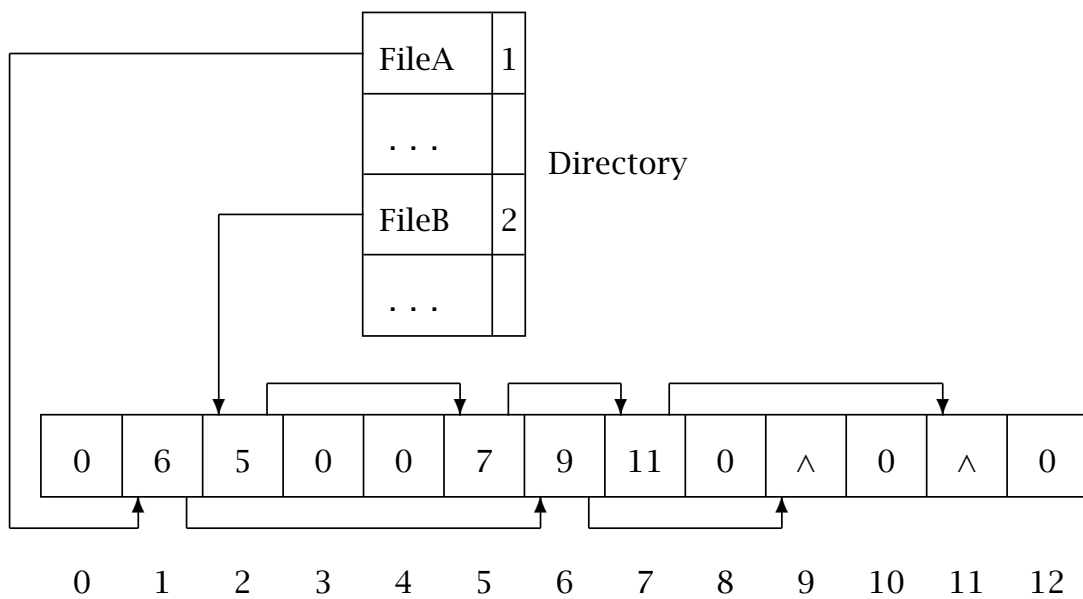


Figure 10.4: Linked allocation

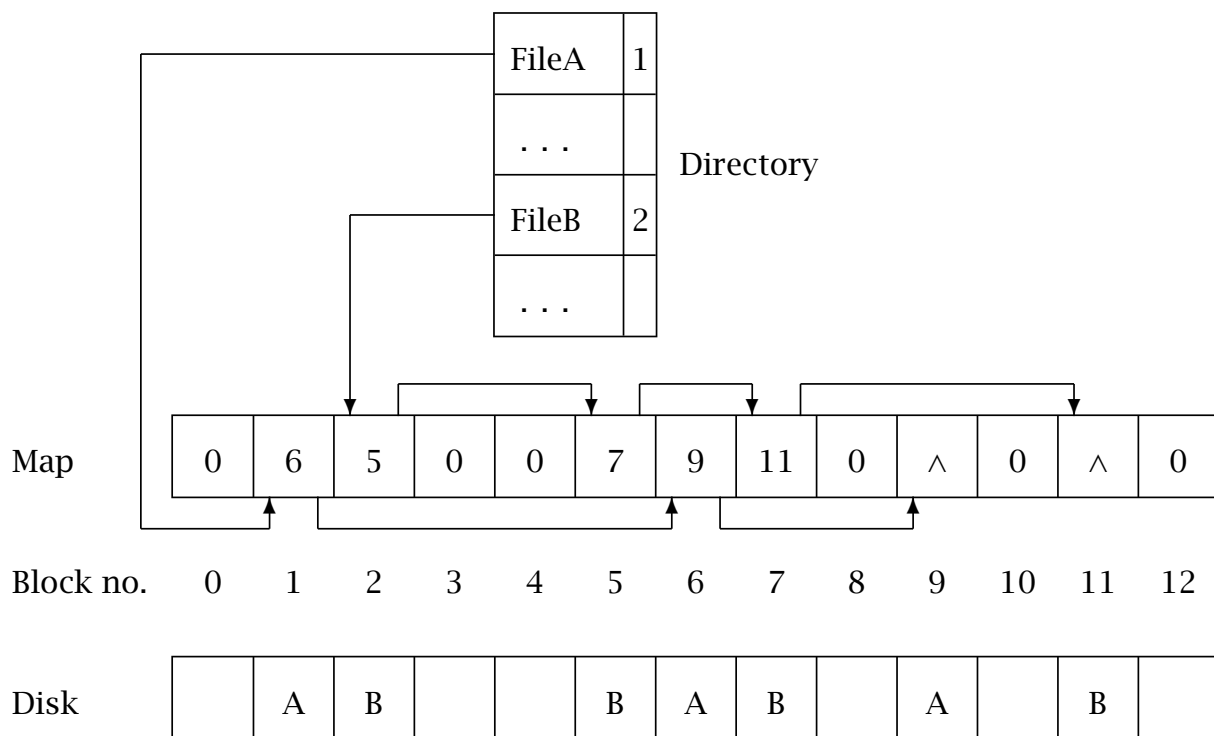


Figure 10.5: Example of a file map

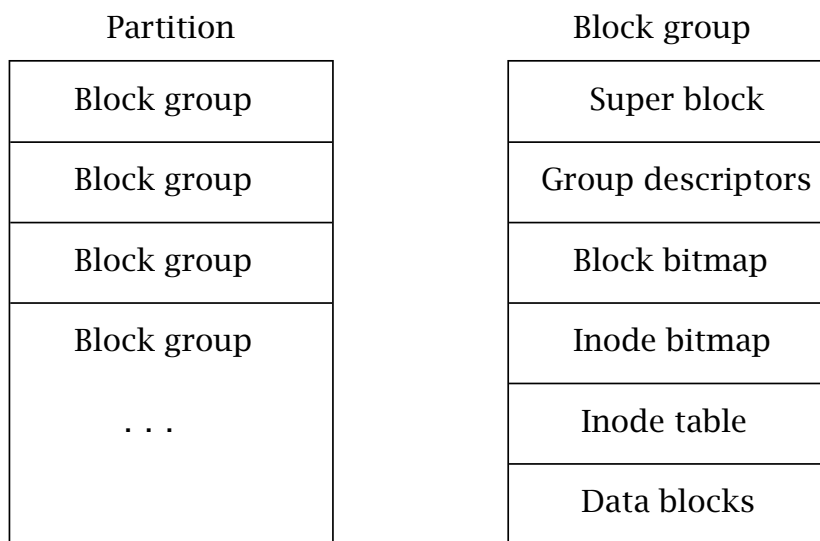


Figure 10.6: Layout of a partition with Ext2

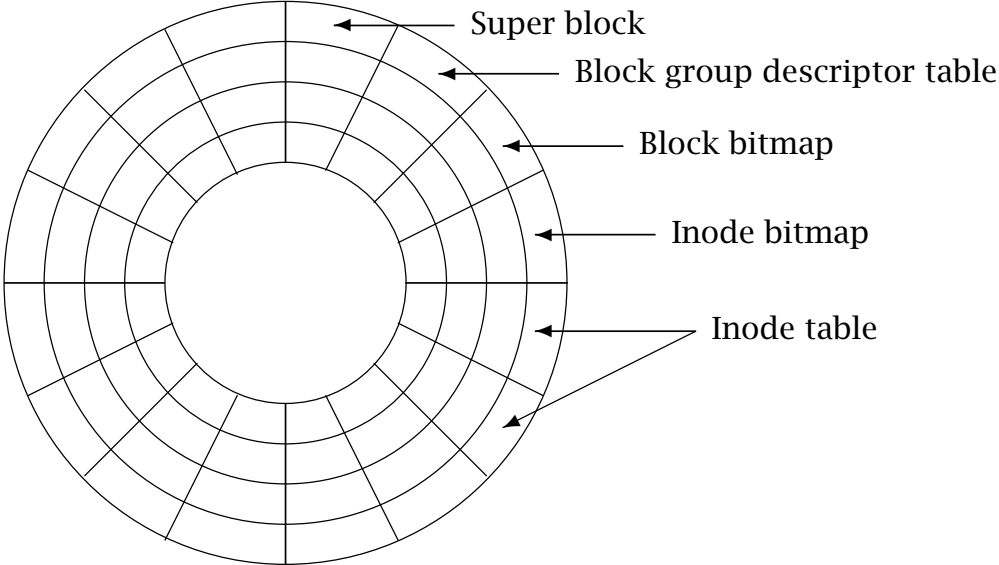


Figure 10.7: A block group on disk

```
struct ext2_super_block{
    u32 s_inodes_count;      /* Inodes count          */
    u32 s_blocks_count;     /* Blocks count         */
    u32 s_free_blocks_count; /* Free blocks count    */
    u32 s_free_inodes_count; /* Free inodes count   */
    u32 s_first_data_block; /* First data block     */
    u32 s_log_block_size;   /* Block size           */
    s32 s_log_frag_size;   /* Fragment size        */
    u32 s_blocks_per_group; /* # Blocks per group   */
    u32 s_inodes_per_group; /* # Inodes per group   */
    u16 s_inode_size;       /* size of inode structure */
    u16 s_block_group_nr;  /* number of this block group */
};
```

Figure 10.8: An Ext2 super block

```
struct ext2_group_desc{
    u32 bg_block_bitmap;      /* Blocks bitmap block */
    u32 bg_inode_bitmap;     /* Inodes bitmap block */
    u32 bg_inode_table;      /* Inodes table block */
    u16 bg_free_blocks_count; /* Free blocks count */
    u16 bg_free_inodes_count; /* Free inodes count */
    u16 bg_used_dirs_count;  /* Directories count */
};
```

Figure 10.9: Structure of a block group descriptor

```
struct ext2_inode{
    u16 i_mode;          /* File mode          */
    u16 i_uid;          /* Owner uid          */
    u32 i_size;         /* Size in bytes      */
    u32 i_atime;       /* Access time        */
    u32 i_ctime;       /* Creation time      */
    u32 i_mtime;       /* Modification time  */
    u16 i_gid;         /* Group id           */
    u16 i_links_count; /* Links count        */
    u32 i_blocks;      /* Blocks count       */
    u32 i_block[15];   /* Pointers to blocks */
    u32 i_faddr;       /* Fragment address   */
    u8  l_i_frag;      /* Fragment number    */
    u8  l_i_fsize;     /* Fragment size      */
};
```

Figure 10.10: Structure of an Ext2 disk inode

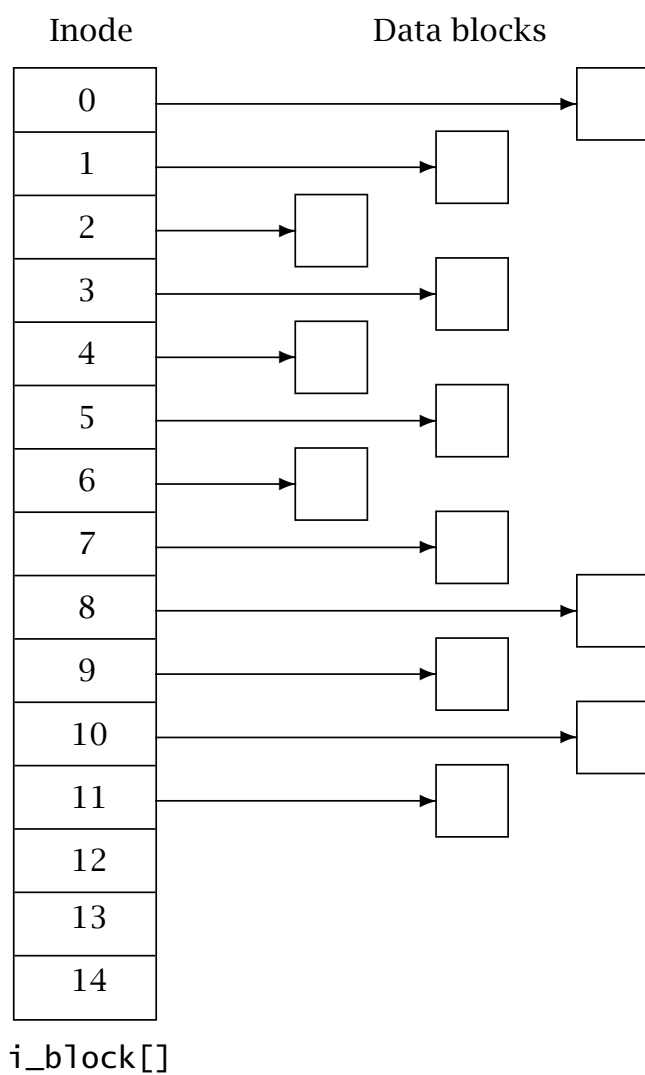


Figure 10.11: Allocation information in an inode

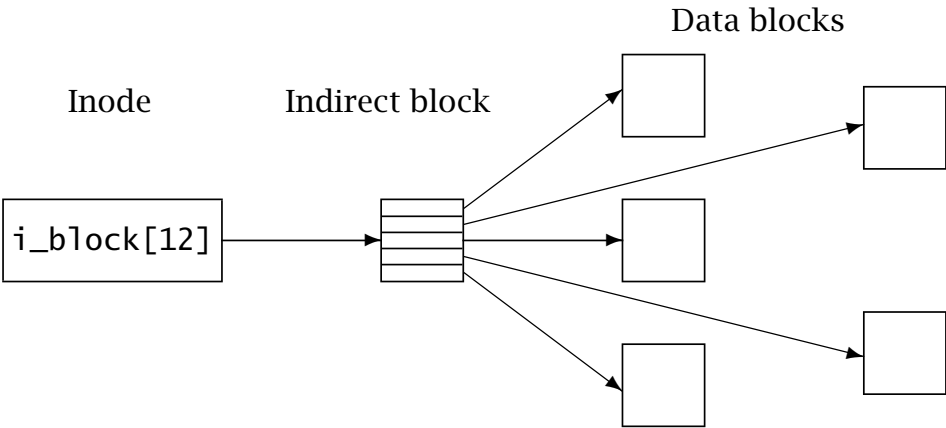


Figure 10.12: Single indirect block

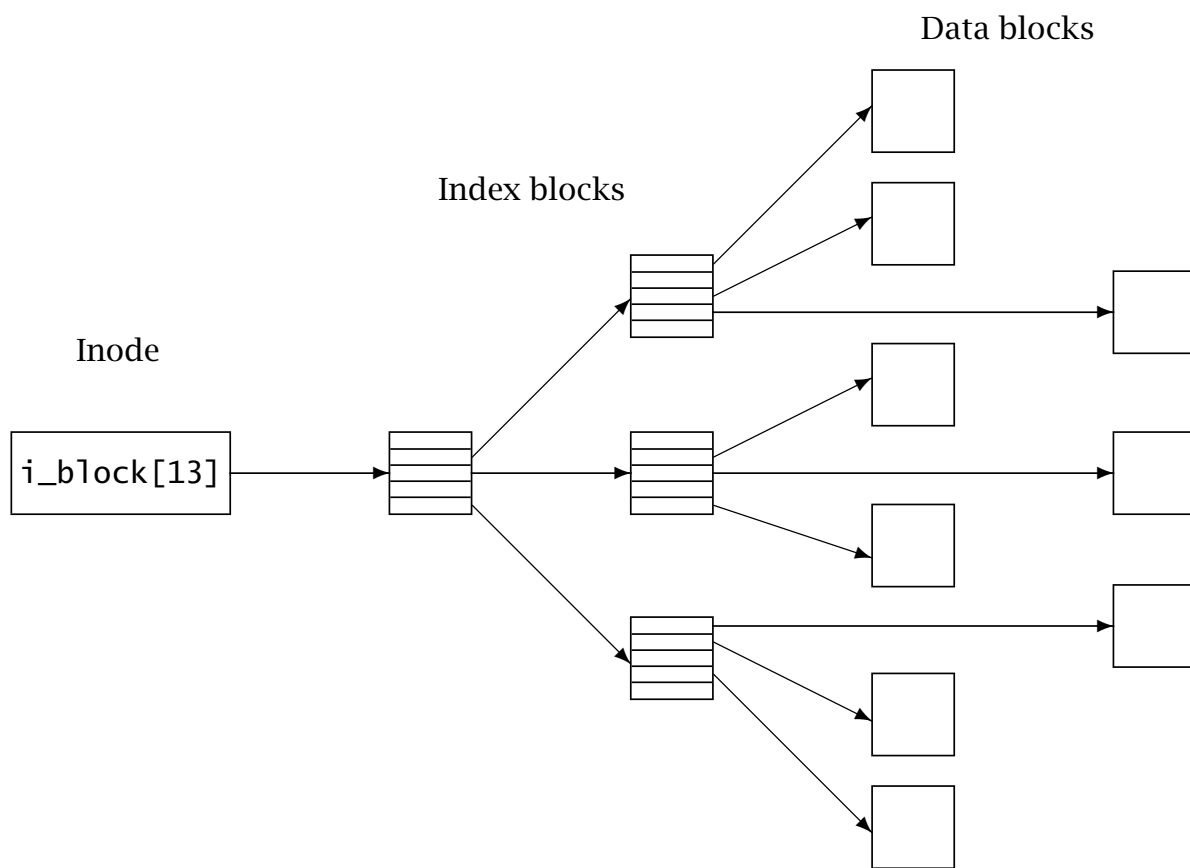


Figure 10.13: Double indirect block

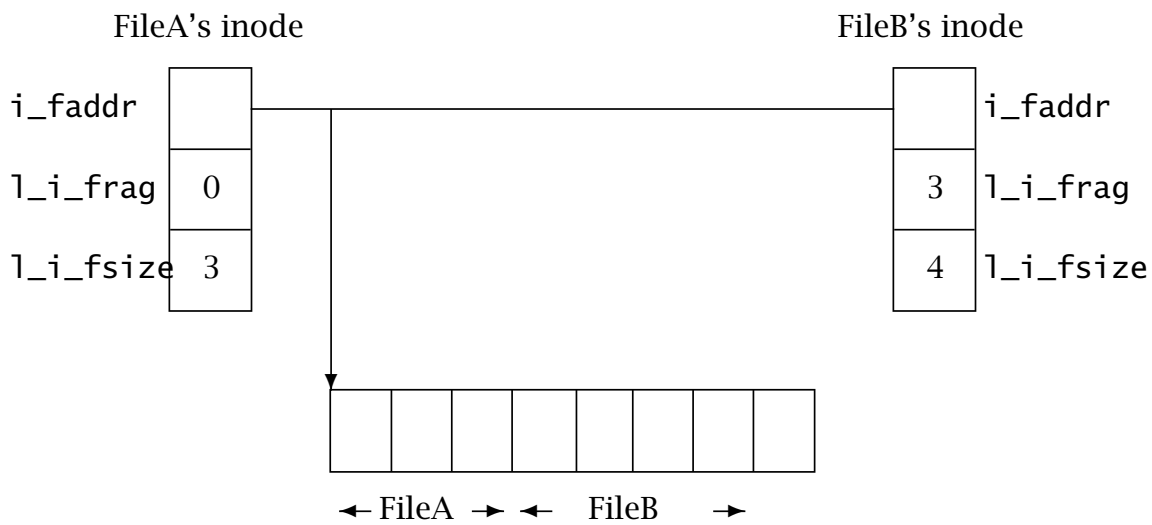


Figure 10.14: Fragments of two files in a block

```
struct ext2_dir_entry{
    u32  inode;    /* inode number of the entry */
    u16  rec_len; /* total length of entry    */
    u16  name_len; /* length of name          */
    char name[]   /* file name               */
};
```

Figure 10.15: An Ext2 directory entry

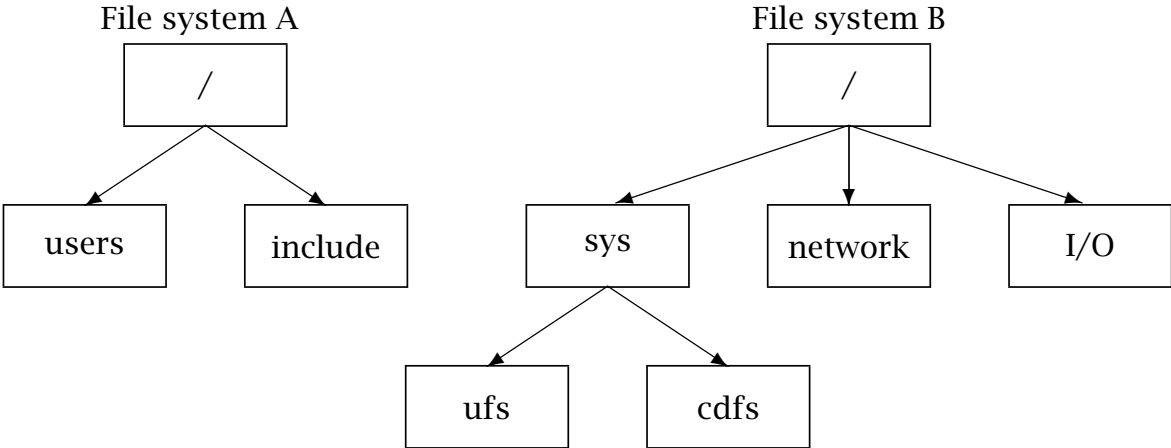


Figure 10.16: File systems as they exist on disk

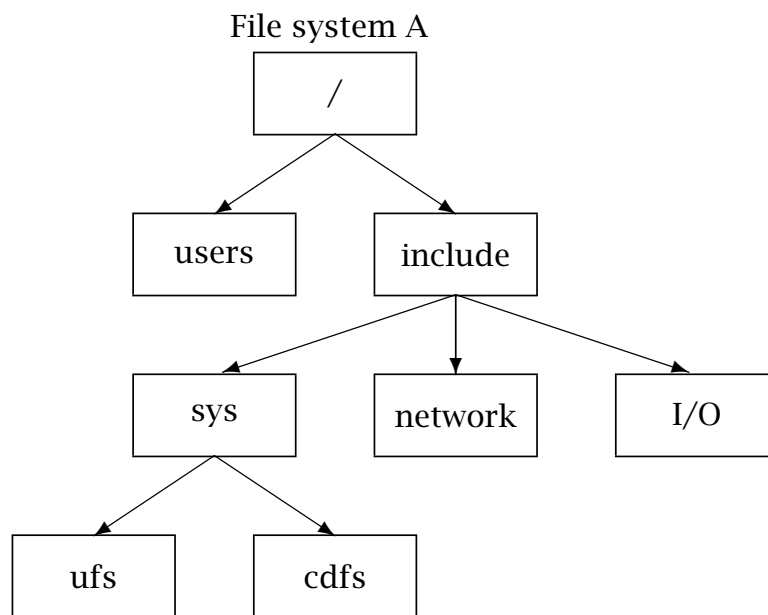


Figure 10.17: Directory structure after attaching file system B

```
struct super_block{
    struct list_head      s_list;
    kdev_t                s_dev;
    struct file_system_type *s_type;
    struct super_operations *s_op;
    struct dentry         *s_root;
    union{
        struct ext2_sb_info  ext2_sb;
        struct msdos_sb_info msdos_sb;
        struct nfs_sb_info   nfs_sb;
    }u;
};
```

Figure 10.18: Data structure representing a mounted file system

```
struct super_operations{
    void (*read_inode)();
    void (*write_inode)();
    void (*put_inode)();
    void (*put_super)();
    void (*write_super)();
    int (*statfs)();
};
```

```
struct ext2_sb_info{
    struct ext2_super_block *s_es;
    struct buffer_head      **s_group_desc;
    struct buffer_head      *s_inode_bitmap[];
    struct buffer_head      *s_block_bitmap[];
};
```

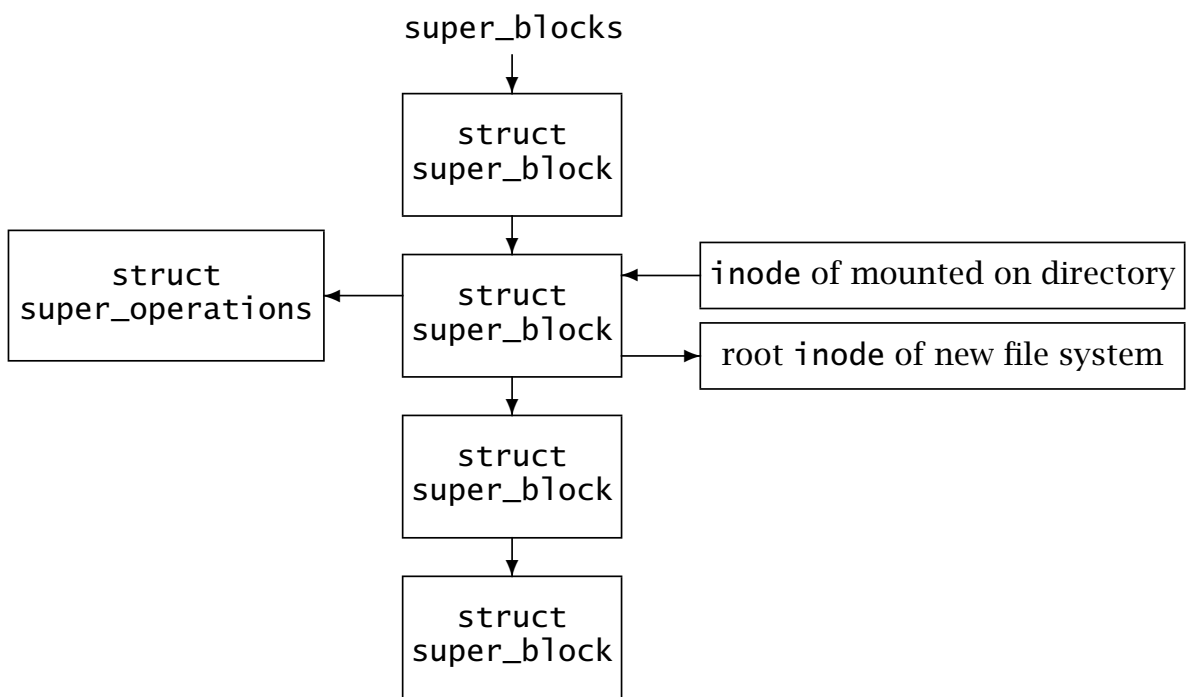


Figure 10.21: Data structures involved in mounting a file system

```
struct ext2_inode_info{
    u32 i_data[15];
    u32 i_faddr;
    u8  i_frag_no;
    u8  i_frag_size;
    u32 i_file_acl;
    u32 i_dir_acl;
    u32 i_block_group;
};
```