



CS3281 / CS5281
Filesystems

CS3281 / CS5281
Spring 2025



Tel (615) 343-7472 | Fax (615) 343-7440
1025 16th Avenue South Nashville, TN 37212
www.isis.vanderbilt.edu



Overview

- A filesystem is an organized collection of files and directories
- The Linux kernel maintains a single hierarchical directory structure to organize all files in the system
 - Not like Windows where each drive (C, D, E, etc) has its own hierarchy
- Root directory is named /
 - Pronounced “slash”

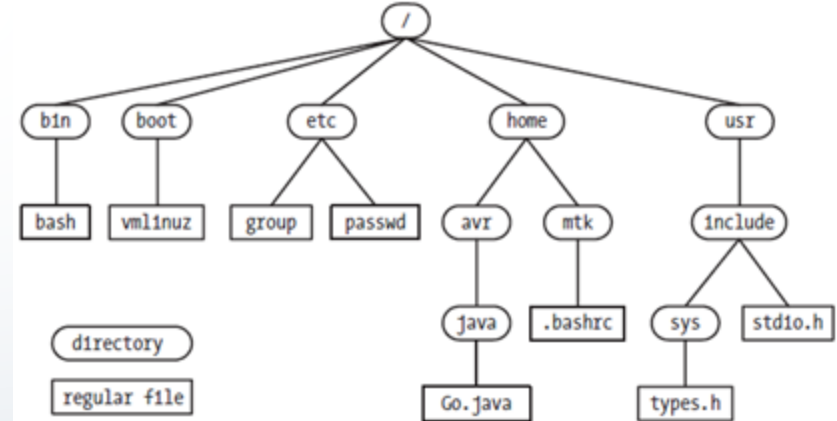


Figure 2-1: Subset of the Linux single directory hierarchy

*Figure from *The Linux Programming Interface* by Michael Kerrisk

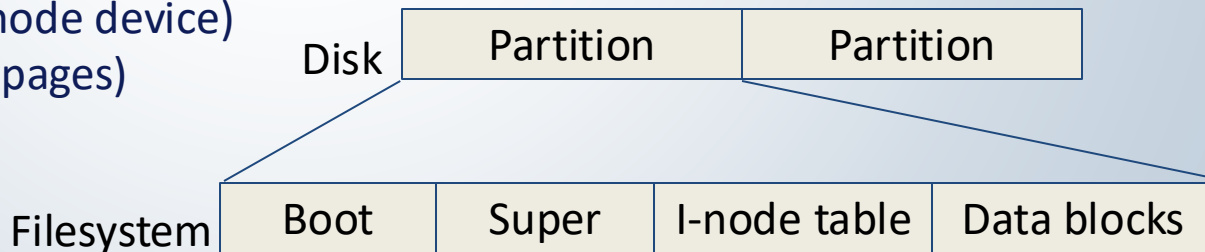
File Types

- Every file has a type
 - This is the character in the first column when you do `ls -l`
- Regular files: ordinary data files, like text files, executables, libraries
- Special files: files other than ordinary data files
 - Devices: represents a device (virtual or physical)
 - Block device (e.g., disk)
 - Character device (keyboard)
 - Named pipes (also called fifos)
 - Directories
 - Symbolic links
- Example on right: block device files

```
daniel@ubuntu:/dev$ ls -l sda*
brw-rw---- 1 root disk 8, 0 Nov  5 09:21 sda
brw-rw---- 1 root disk 8, 1 Nov  5 09:21 sda1
daniel@ubuntu:/dev$
```

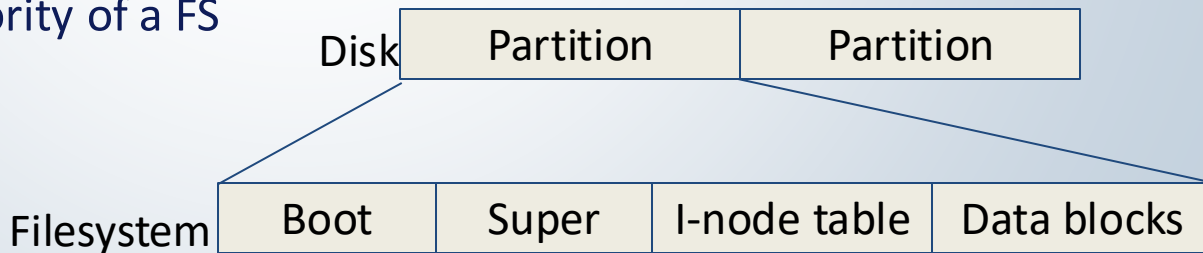
Back to Filesystems

- A disk drive is divided into circles called tracks
 - Tracks are divided into sectors
 - Sectors are a series of physical blocks
 - Physical block: the smallest unit a disk can read or write
 - Usually 512 bytes (older disks) or 4096 bytes (newer disks)
- Each disk is divided into partitions
 - Each is a separate device under /dev
 - A partition holds either
 - *Filesystem* (on-disk structures)
 - Data area (raw-mode device)
 - Swap (for virtual pages)



Filesystem Structure

- Boot block: always the first block in a filesystem (FS)
 - Not used by FS; contains info to boot the OS
 - Only one needed by OS
- Super block: contains parameter info about the filesystem
 - Size of the i-node table, size of logical blocks, size of the filesystem (in logical blocks)
- I-node table: contains one (unique) entry for every file in file system
 - Contains most of the “metadata” about individual file
- Data blocks: the (logical) blocks that contain the data for files and directories
 - This is the vast majority of a FS



I-Nodes

- Index nodes (i-nodes) contains the following metadata about a file
 - File type (for example, regular, char device, block device, directory, symbolic link)
 - Owner of the file
 - Group of the file
 - File access permissions for three categories: user (owner), group, other
 - Three timestamps:
 - Time of last access (`ls -lu`)
 - Time of last modification (default timestamp in `ls -l`)
 - Time of last status change (change to i-node info) (`ls -lc`)
 - Number of hard links (pathnames) to file
 - Size of the file (in bytes)
- Number of blocks allocated to file
 - Pointers to the data blocks

Directories

- A directory is stored in a filesystem in a similar way as a regular file, but
 - It is marked as a directory in its i-node
 - It's a file with a special organization: it's a table consisting of filenames and i-node numbers
- Example is on the right
- Note: the i-node doesn't have a filename!
 - Implication: you can have multiple links to the same file!

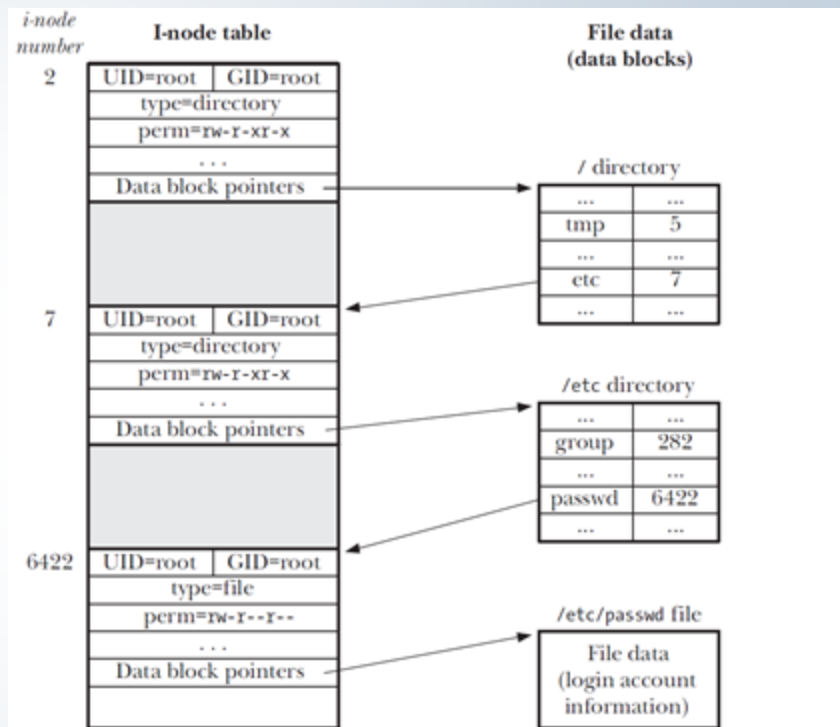


Figure 18-1: Relationship between i-node and directory structures for the file `/etc/passwd`

*Figure from *The Linux Programming Interface* by Michael Kerrisk

Data blocks

- How can files of very different sizes be supported?
 - One method: store pointers to the data blocks!
- Figure on the right shows how ext2 does this
 - Small files might fit entirely in direct pointers
- Bigger files use:
 - Indirect pointers
 - Double-indirect pointers
 - Triple-indirect pointers
- Advantages of pointers
 - Fixed-size i-node
 - But arbitrary size files
 - Store blocks non-contiguously

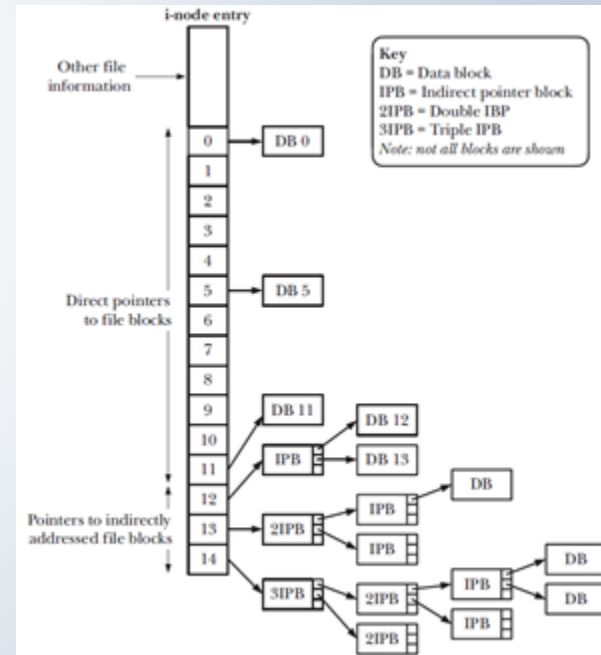
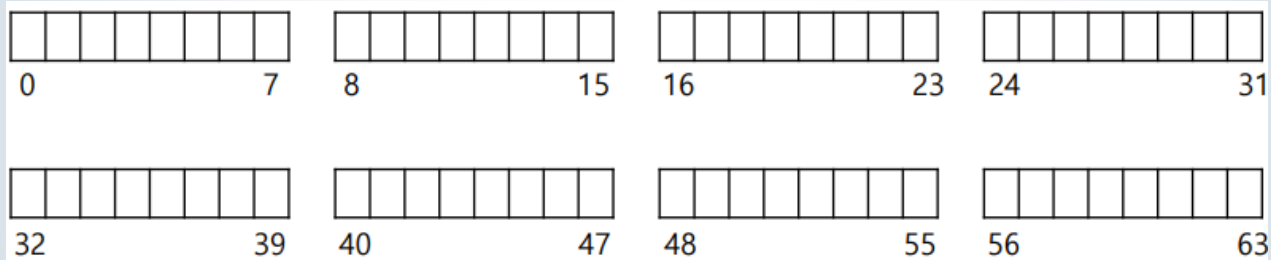


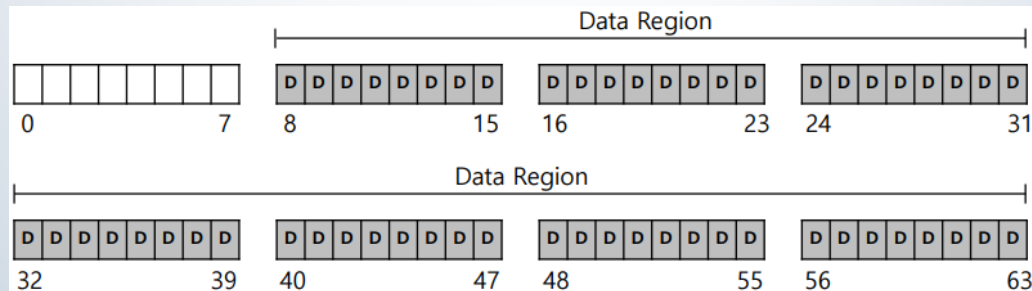
Figure 14-2: Structure of file blocks for a file in an ext2 file system

Very Simple File System (VSFS) Data Structures

- Divide disk into blocks
- Use one block size (4KB)
- Blocks are addressed from 0 to $N-1$ (N is the number of blocks)



- Store user data in *data region* (e.g., files and directories)



The I-Node

- Index node (inode): array of nodes is indexed
- Each inode is identified by an i-number
 - Used for indexing an array of inodes
- Find the byte address for the inode with i-number 32
 - Compute the offset into the inode table: $32 * \text{sizeof}(\text{inode}) = 32 * 256 = 8192$ (8KB)
 - Add the offset to start address of inode table: $12\text{KB} + 8\text{KB} = 20\text{KB}$
- inodes are fetched using *sectors* (a block consists of sectors)
 - 512-byte sectors
 - Sector number: $(20 * 1024)/512 = 40$

The Inode table																																			
				iblock 0				iblock 1				iblock 2				iblock 3				iblock 4															
Super	i-bmap	d-bmap	0	1	2	3	16	17	18	19	32	33	34	35	48	49	50	51	64	65	66	67													
			4	5	6	7	20	21	22	23	36	37	38	39	52	53	54	55	68	69	70	71													
			8	9	10	11	24	25	26	27	40	41	42	43	56	57	58	59	72	73	74	75													
			12	13	14	15	28	29	30	31	44	45	46	47	60	61	62	63	76	77	78	79													
0KB				4KB				8KB				12KB				16KB				20KB				24KB				28KB				32KB			