file management

File Management

- File management system consists of system utility software that run as privileged applications
  - usually part of the os kernel
  - usually based on block-oriented devices (i.e. disks)

File System Properties

- Structure
- Long-term existence
- Sharable between processes

File Operations

- Create
- Delete
- Open
- Close
- Read
- Write

Terms Used with Files

- Field
  - Basic element of data
  - Contains a single value
  - Characterized by its length and data type
- Record
  - Collection of related fields
  - Treated as a unit
    - Example: employee record

Terms Used with Files

- File
  - Collection of similar records
  - Treated as a single entity
  - Have file names
  - May restrict access
- Database
  - Collection of related data
  - Relationships exist among elements

Typical Operations

- Retrieve_All
- Retrieve_One
- Retrieve_Next
- Retrieve_Previous
- Insert_One
- Delete_One
- Update_One
- Retrieve_Few

Objectives for a File Management System

- Meet the data management needs and requirements of the user
- Guarantee that the data in the file are valid
- Optimize performance
- Provide I/O support for a variety of storage device types

Objectives for a File Management System

- Minimize or eliminate the potential for lost or destroyed data
- Provide a standardized set of I/O interface routines
- Provide I/O support for multiple users

file systems need data structures

- to keep track of which blocks are allocated to a given file
- to keep track of free blocks
- to keep names, directories, access rights, etc

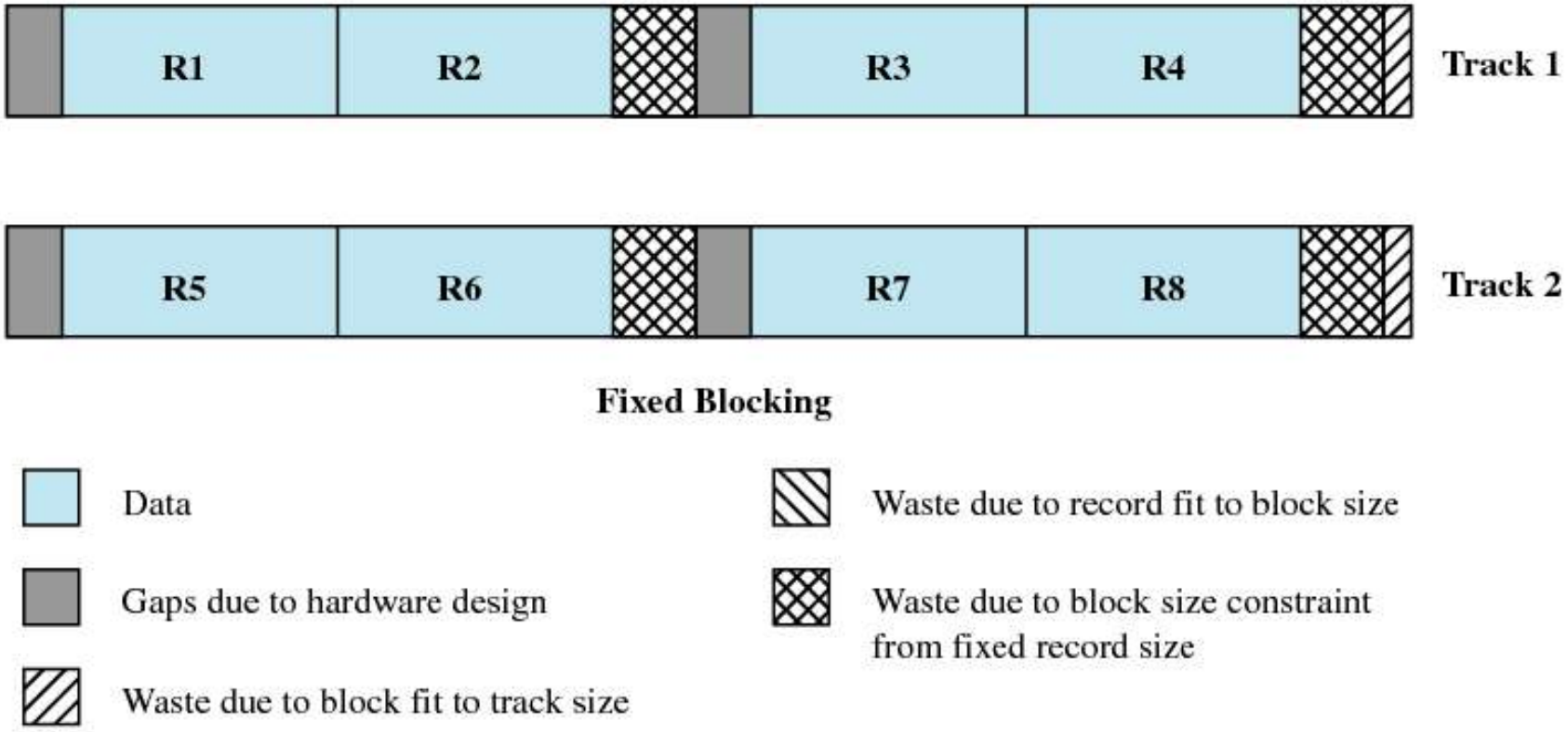
Criteria for File Organization

- Short access time
  - Needed when accessing a single record
  - Not needed for batch mode
- Ease of update
  - File on CD-ROM will not be updated, so this is not a concern

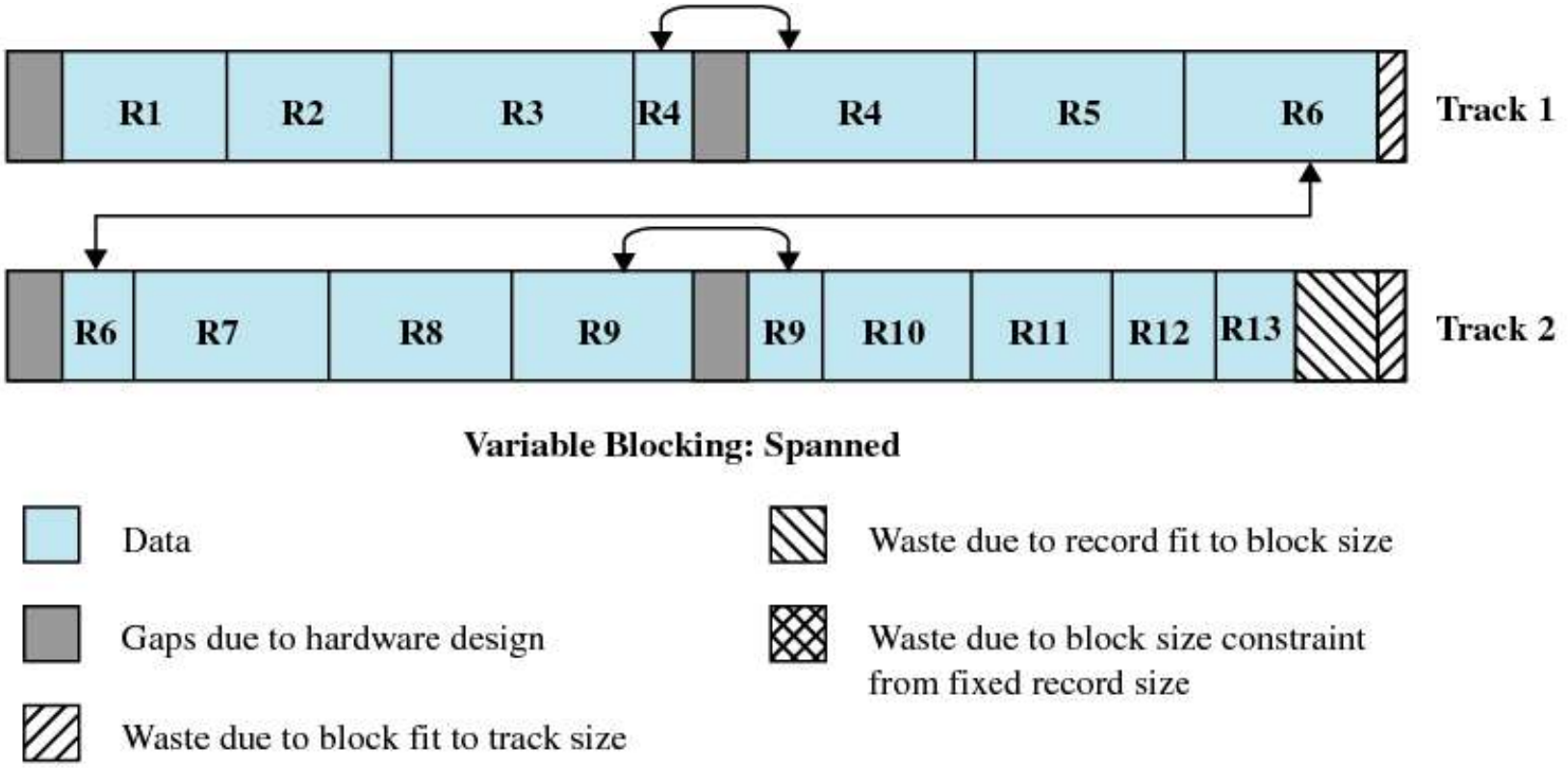
Criteria for File Organization

- Economy of storage
  - Should be minimum redundancy in the data
  - Redundancy can be used to speed access such as an index
- Simple maintenance
- Reliability

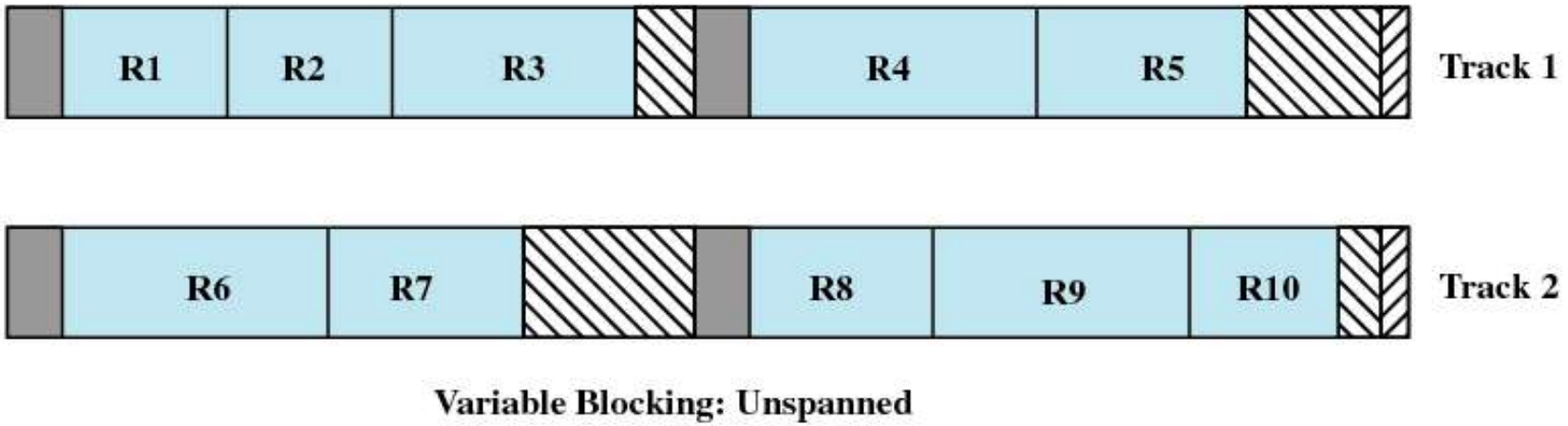
Fixed Blocking



Variable Blocking: Spanned



Variable Blocking Unspanned



Secondary Storage Management

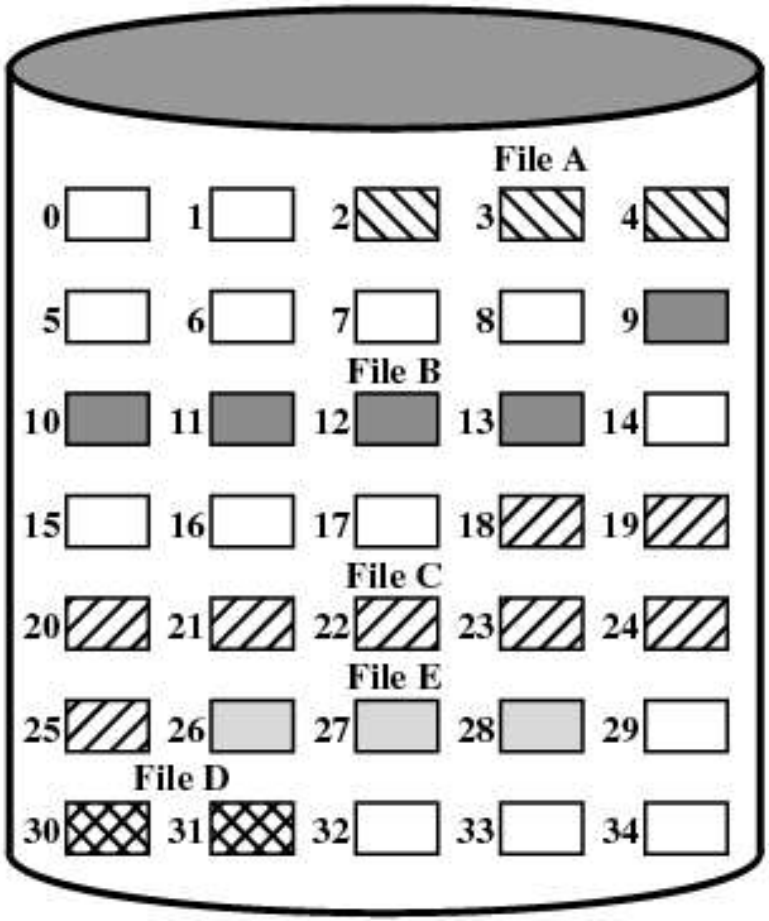
- Space must be allocated to files
- Must keep track of the space available for allocation
- bitmaps
- free ranges

Preallocation

- Need the maximum size for the file at the time of creation
- Difficult to reliably estimate the maximum potential size of the file
- Tend to overestimated file size so as not to run out of space

Methods of File Allocation

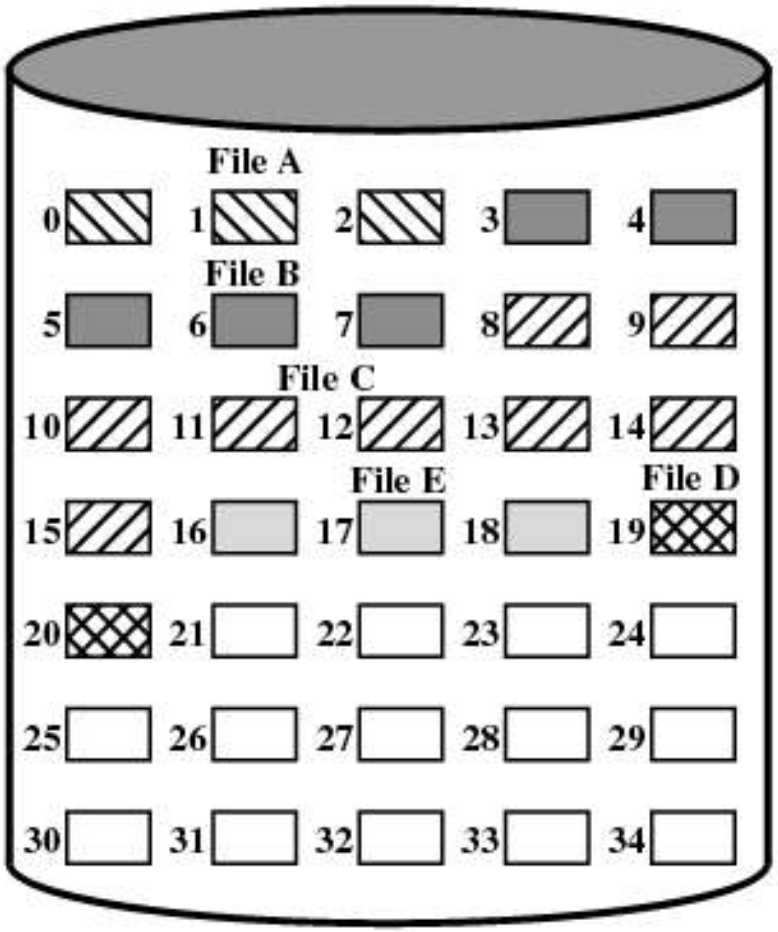
- Contiguous allocation
  - Single set of blocks is allocated to a file at the time of creation
  - Only a single entry in the file allocation table
    - Starting block and length of the file
  - External fragmentation will occur
    - best fit, first fit, nearest fit
    - Need to perform compaction



File Allocation Table

File Name	Start Block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3

Figure 12.7 Contiguous File Allocation



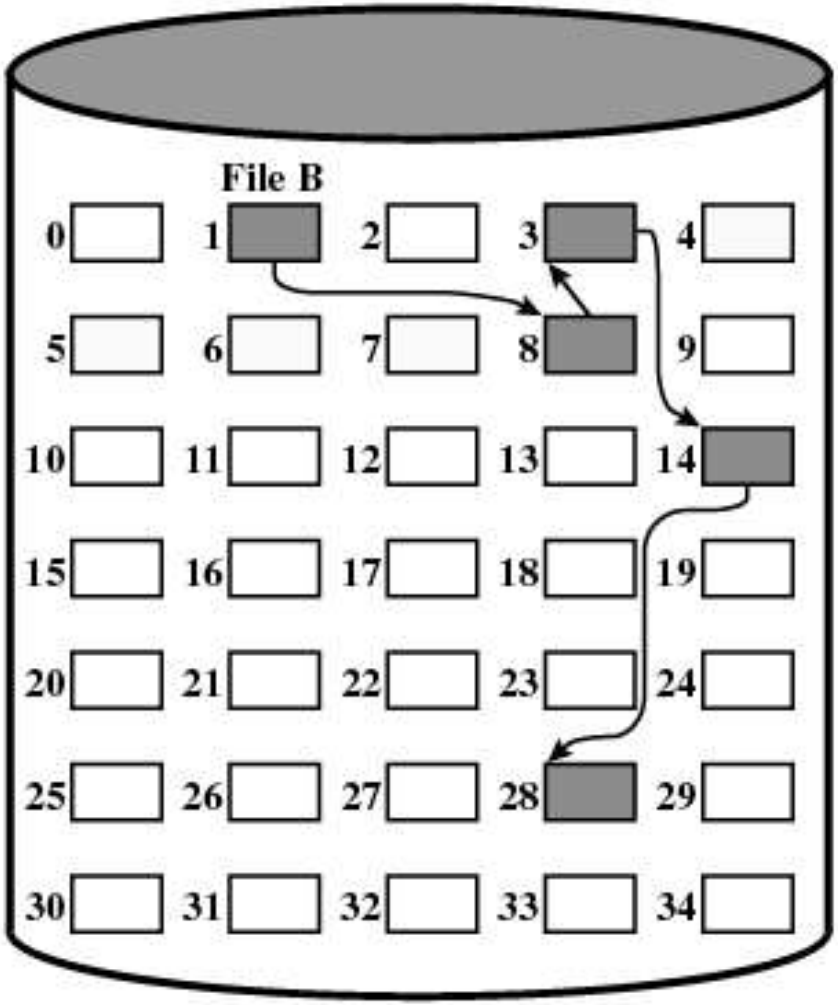
File Allocation Table

File Name	Start Block	Length
File A	0	3
File B	3	5
File C	8	8
File D	19	2
File E	16	3

Figure 12.8 Contiguous File Allocation (After Compaction)

Methods of File Allocation

- Chained allocation
  - Allocation on basis of individual block
  - FAT contains a pointer to the first block
  - Each block contains a pointer to the next block in the chain
  - Only single entry in the file allocation table
    - Starting block and length of file
- No external fragmentation
- No FAT read overhead for sequential files
- No accommodation of the principle of locality



File Allocation Table

File Name	Start Block	Length
...	...	...
File B	1	5
...	...	...

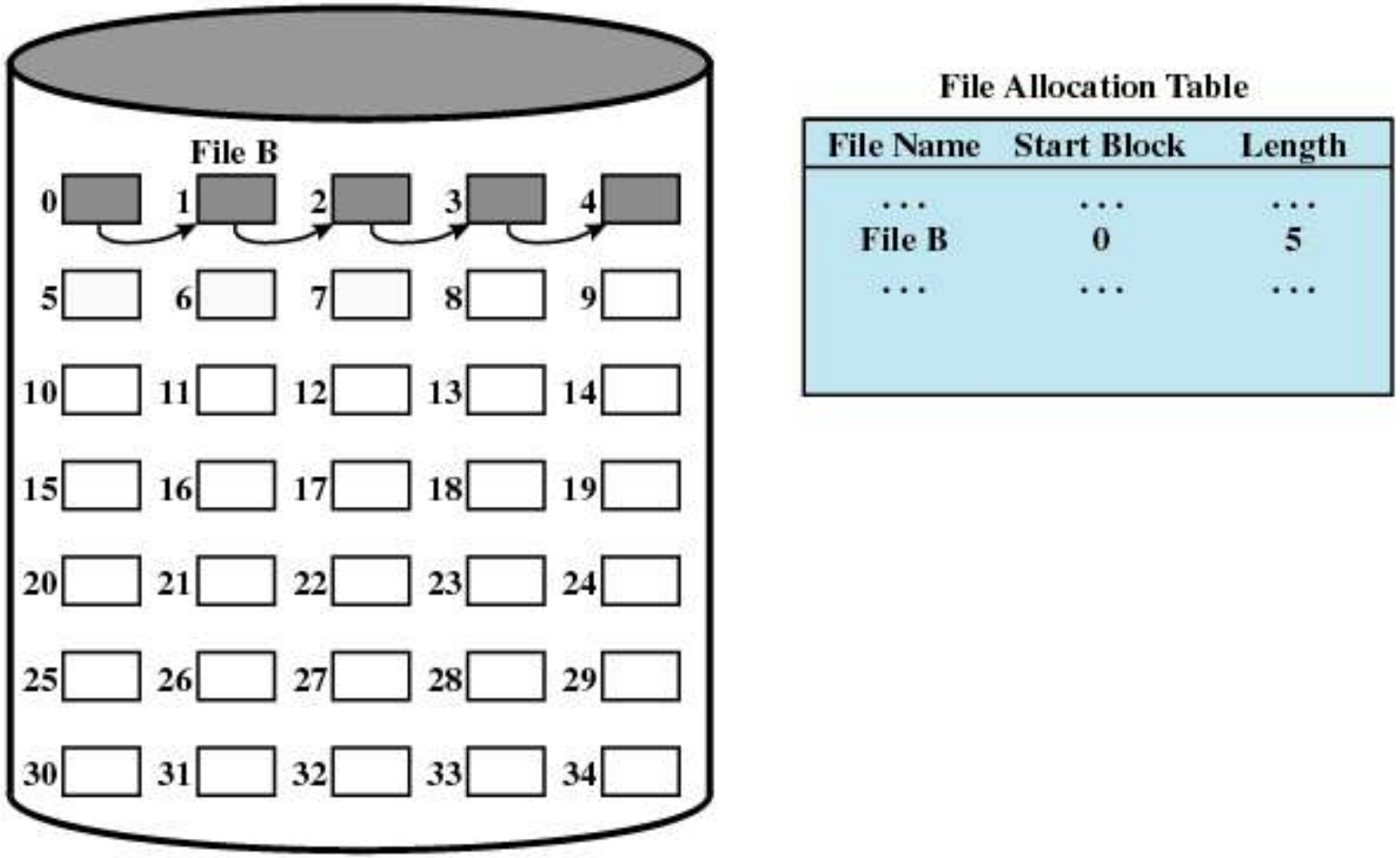


Figure 12.10 Chained Allocation (After Consolidation)

Methods of File Allocation

- Indexed allocation
  - File allocation table contains a separate one-level index for each file
  - The index has one entry for each portion allocated to the file
  - The file allocation table contains block number for the index

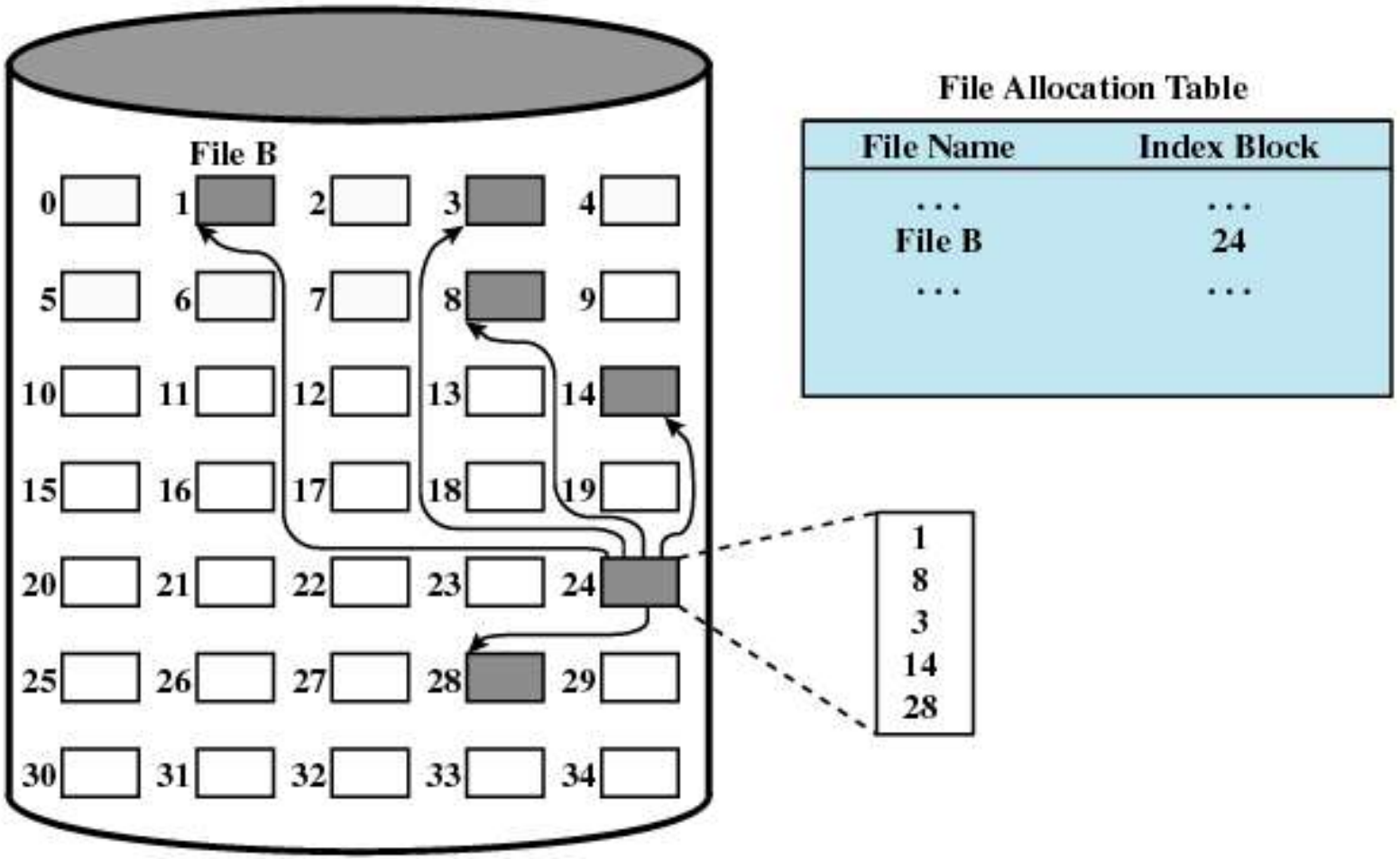


Figure 12.11 Indexed Allocation with Block Portions

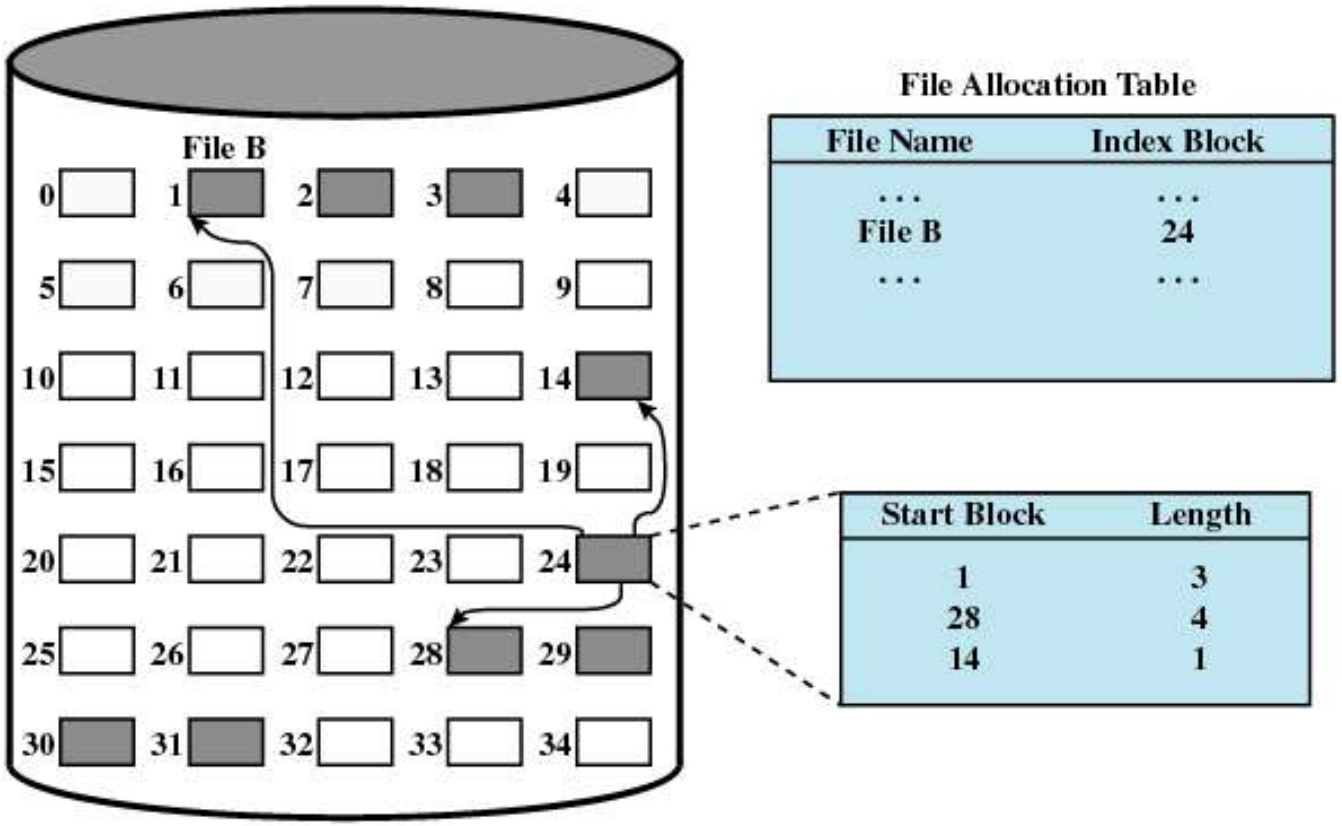


Figure 12.12 Indexed Allocation with Variable-Length Portions

UNIX File Management

- Types of files
  - Regular, or ordinary
    - a sequence of bytes (no records!)
  - Directory
  - Special (character of block devices)
  - Named pipes (FIFO)
  - Links (hard links)
  - Symbolic links (soft links)

Inodes

- Index node
- Control structure that contains key information for a particular file

File Mode	16-bit flag that stores access and execution permissions associated with the file. 12-14 File type (regular, directory, character or block special, FIFO pipe 9-11 Execution flags 8 Owner read permission 7 Owner write permission 6 Owner execute permission 5 Group read permission 4 Group write permission 3 Group execute permission 2 Other read permission 1 Other write permission 0 Other execute permission
Link Count	Number of directory references to this inode
Owner ID	Individual owner of file
Group ID	Group owner associated with this file
File Size	Number of bytes in file
File Addresses	39 bytes of address information
Last Accessed	Time of last file access
Last Modified	Time of last file modification
Inode Modified	Time of last inode modification

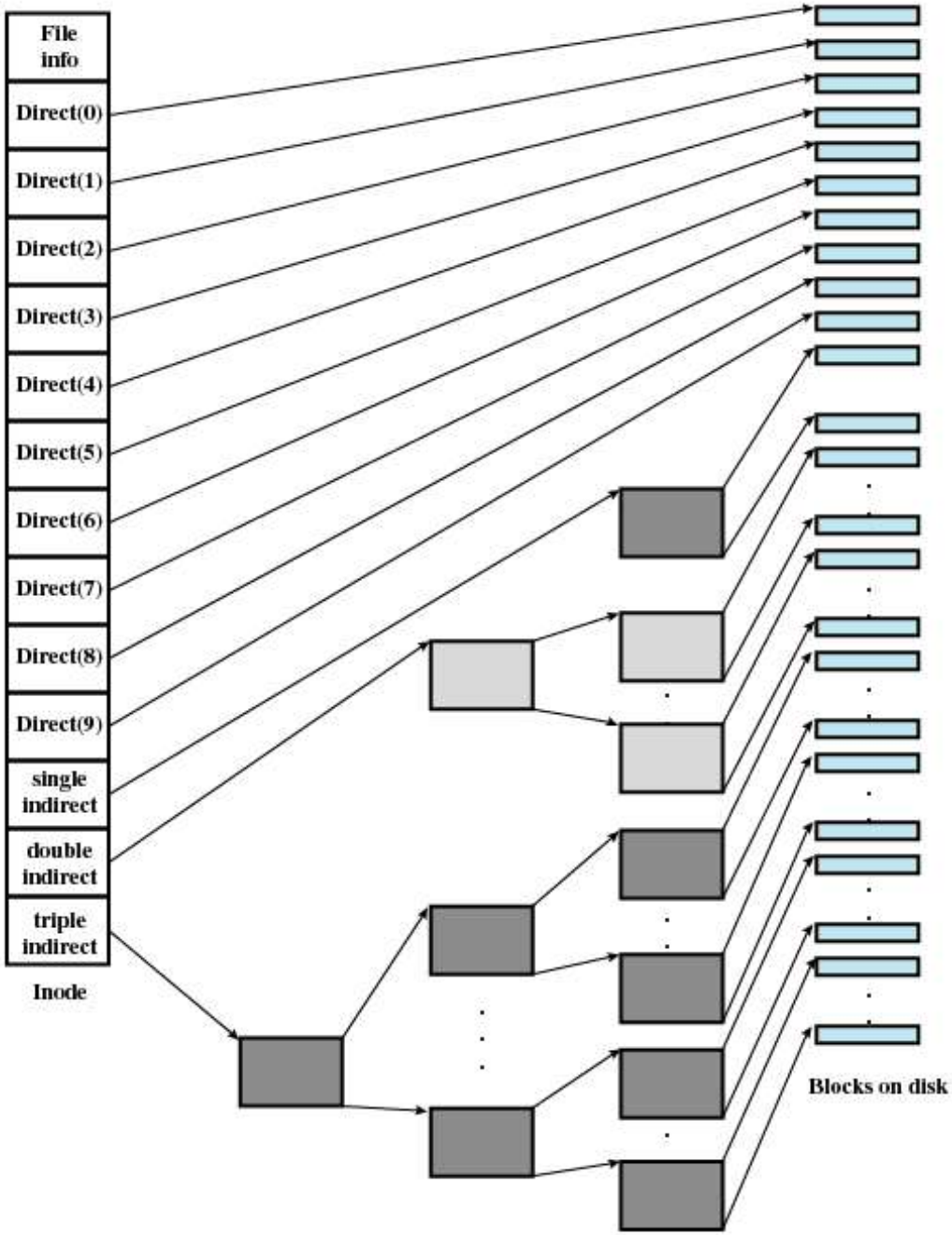


Figure 12.13 Layout of a UNIX File on Disk

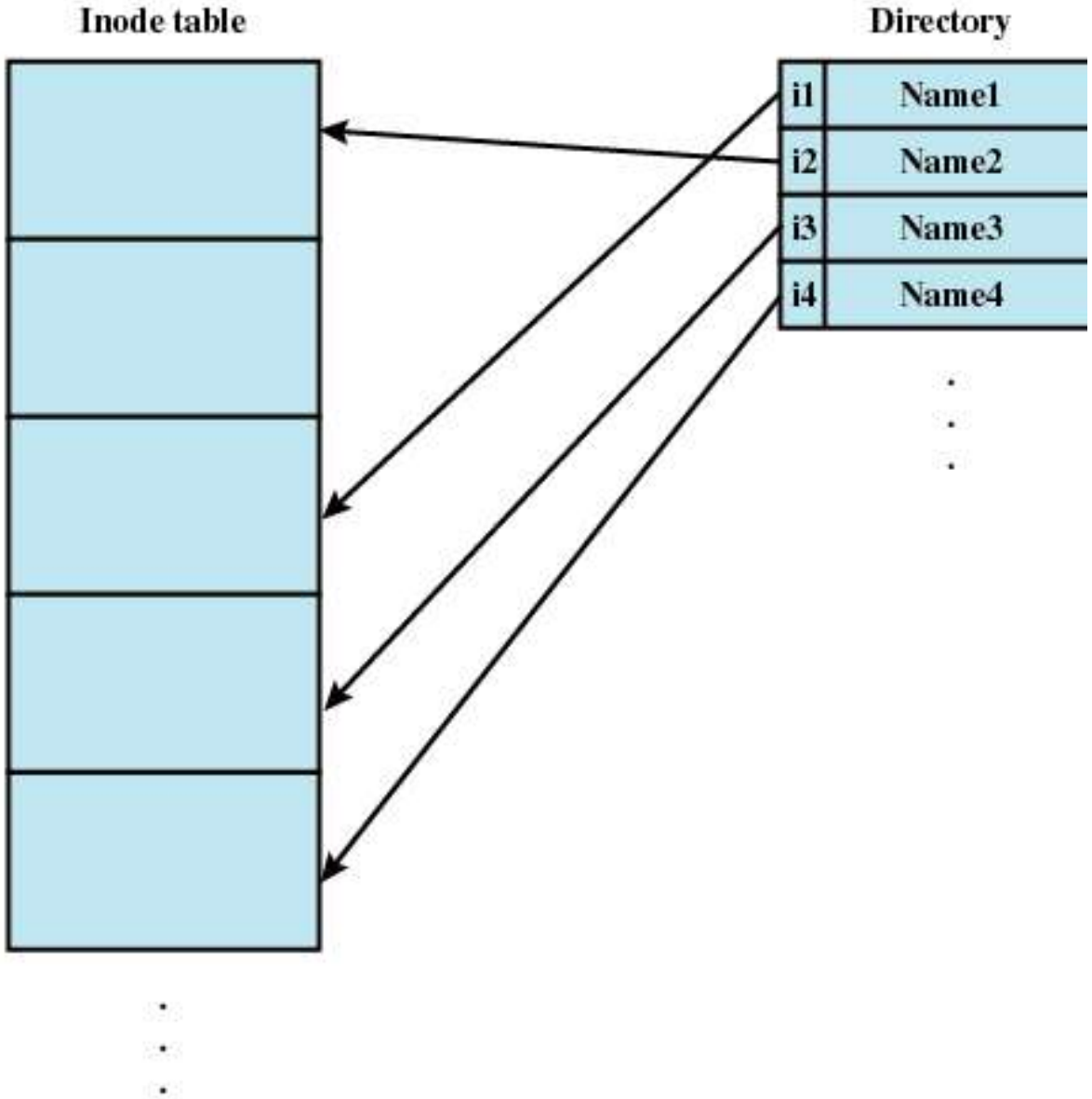


Figure 12.14 UNIX Directories and Inodes

Linux Virtual File System

- Uniform file system interface to user processes
- Represents any conceivable file system's general feature and behavior
- Assumes files are objects that share basic properties regardless of the target file system

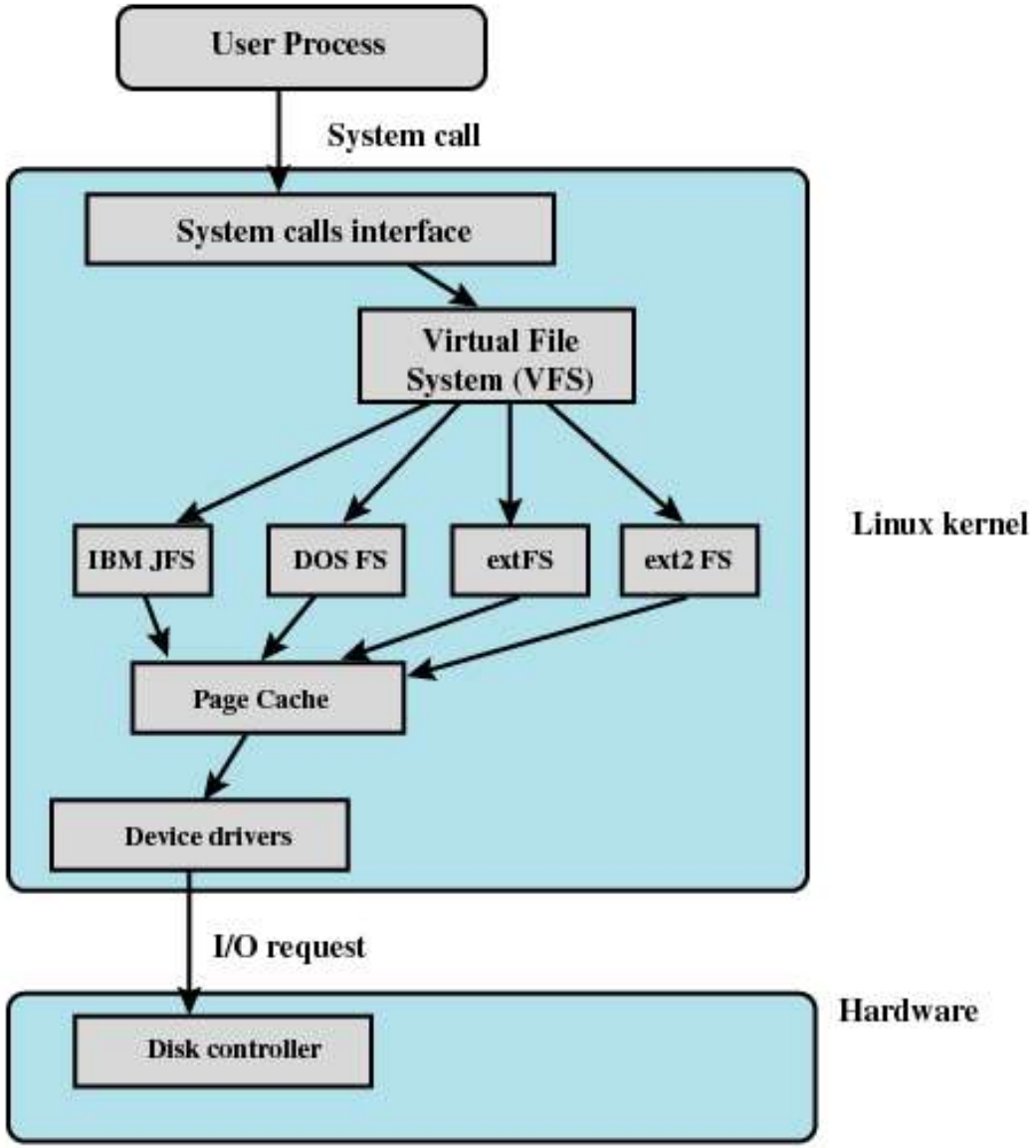
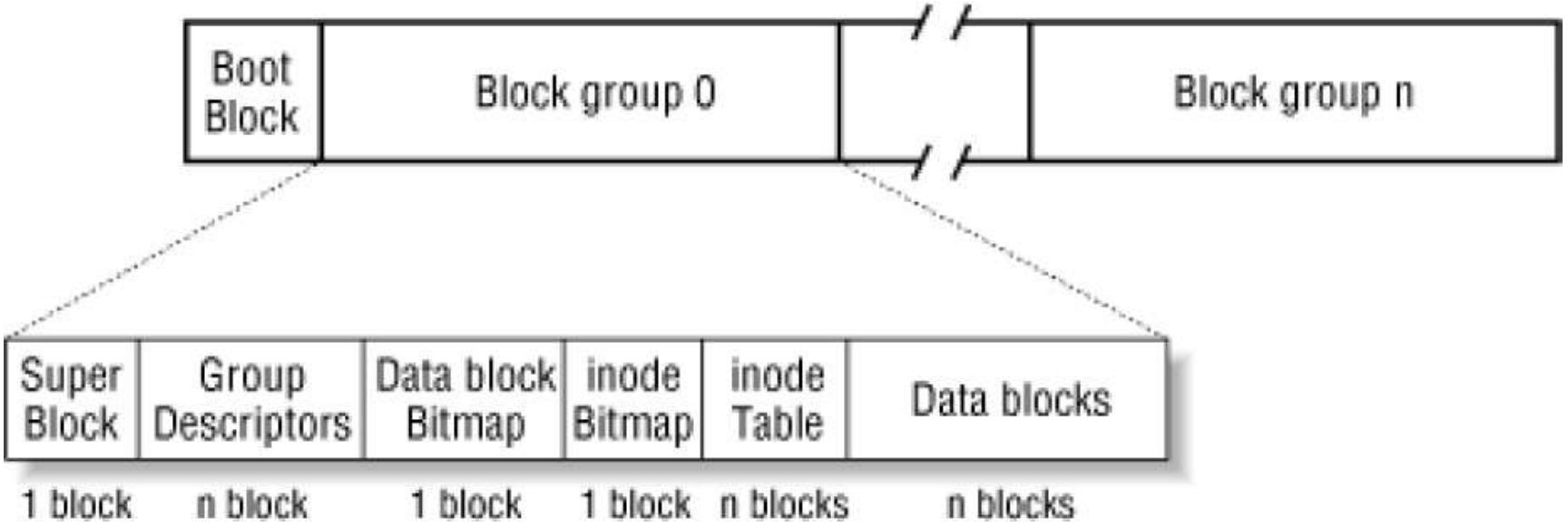


Figure 12.15 Linux Virtual File System Context

Primary Objects in VFS

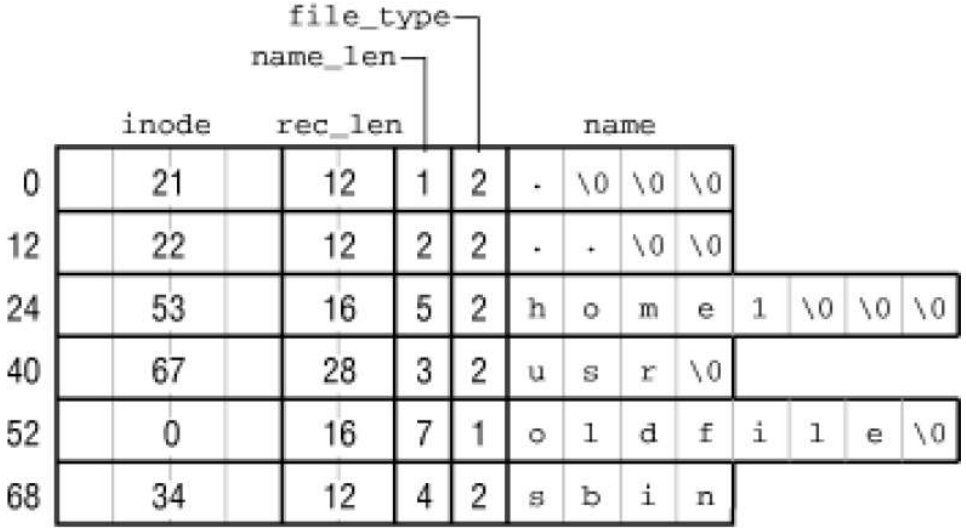
- Superblock object
  - Represents a specific mounted file system
- Inode object
  - Represents a specific file
- Dentry object
  - Represents a specific directory entry
- File object
  - Represents an open file associated with a process

ext2 file system



- for each group
  - all groups are of the same length
  - each group says where all other groups are (redundancy)
  - 2 bitmaps for free/allocated spaces
    - both data blocks and inodes of this group

ext2 file system



- directory stored in variable record length format
- stored as a file
- given the inode *i* number where is it stored on the disk?

  group: *i* / inodes_in_one_group

  inode: i % inodes_in_one_group