Chapter 12 – File management

Fontys › FOR SOCIETY

Roadmap

- Overview
- File organisation and Access
- File Directories
- File Sharing
- Record Blocking
- Secondary Storage Management
- File System Security
- Unix File Management
- Linux Virtual File System
- Windows File System

Files

- Files are the central element to most applications
- The File System is one of the most important part of the OS to a user
- Desirable properties of files:
  - Long-term existence
  - Sharable between processes
  - Structure (to serve needs of apps)

Files systems

- Provide a means to store data organized as files as well as a collection of functions that can be performed on files
- Maintain a set of attributes associated with the file
- Typical operations:
  - Create
  - Delete
  - Open
  - Close
  - Read
  - Write

Database
File Structure Terms

Four terms are common when discussing 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
- File
- Database

Typical File Operations on Record level

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

…"record(s)"

File structure terms (2)

- File (=Table)
  - Have file names
  - Is a collection of similar records
  - Treated as a single entity
  - May implement access control mechanisms
- Database
  - Collection of related data
  - Relationships exist among elements
  - Consists of one or more files

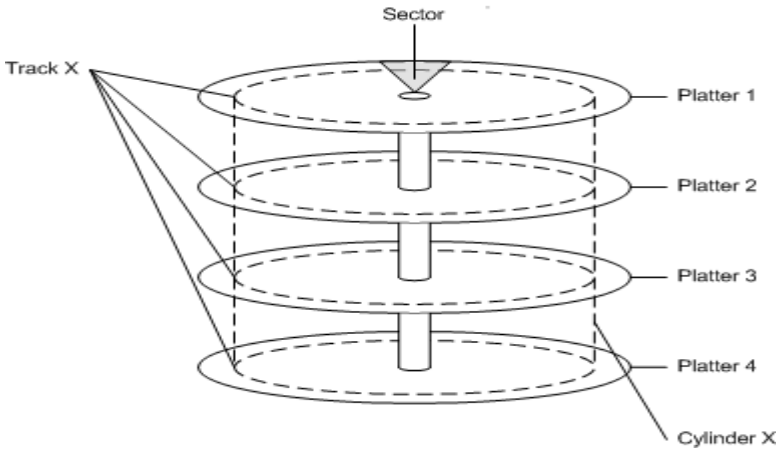
Requirements for a general purpose FMS(1/2)

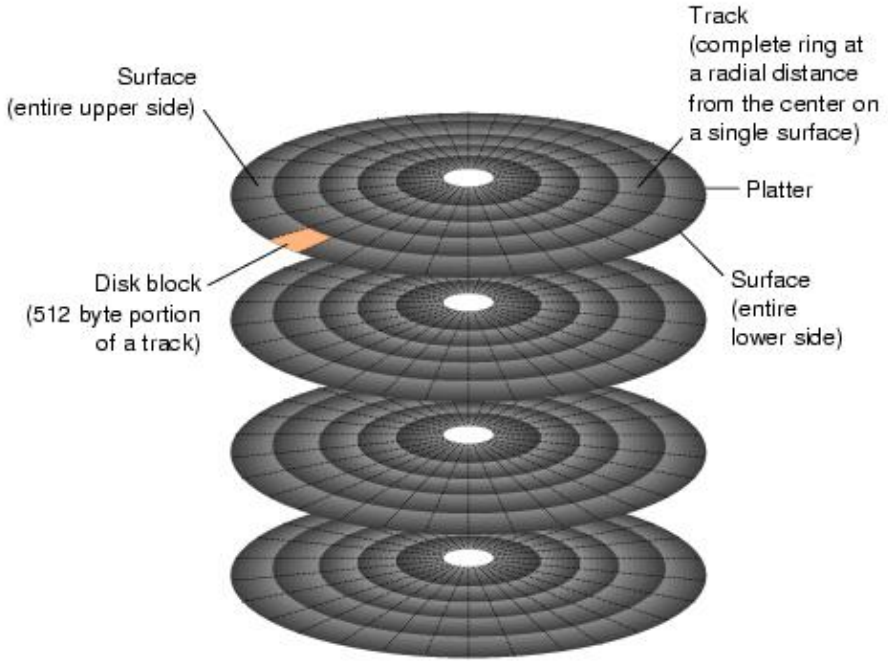
1. Each user should be able to create, delete, read, write and modify files
2. Each user may have controlled access to other users' files
3. Each user may control what type of accesses are allowed to the users' files
4. Each user should be able to restructure the user's files in a form appropriate to the problem

Requirements for a general purpose FMS(2/2)

5. Each user should be able to move data between files
6. Each user should be able to back up and recover the user's files in case of damage
7. Each user should be able to access the user's files by using symbolic names

Block-Track-sector-platter





Fontys

Typical software organization

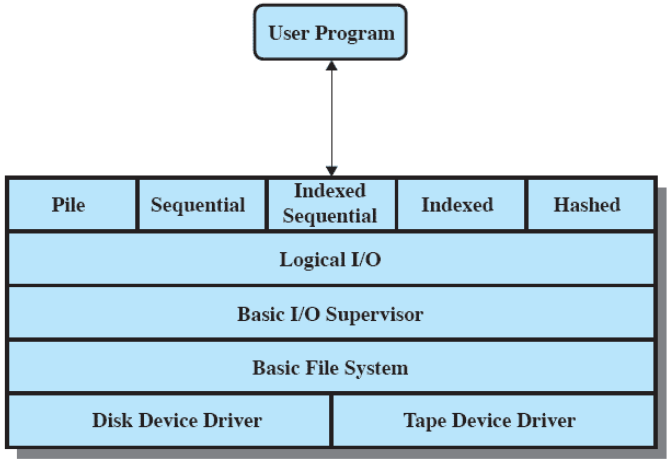


Figure 12.1 File System Software Architecture

Device Drivers

- Lowest level
- Communicates directly with peripheral devices
- Responsible for starting **I/O** operations on a device
- Processes the completion of an **I/O** request

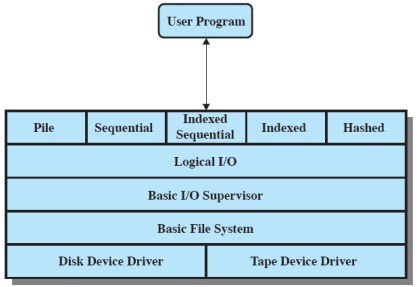


Figure 12.1 File System Software Architecture

Basic File System

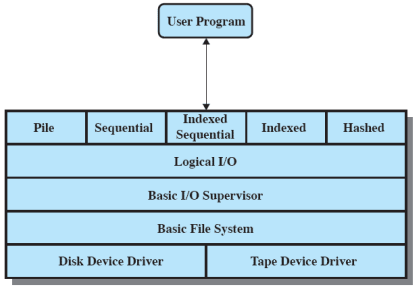


Figure 12.1 File System Software Architecture

- Also referred as the Physical I/O level
- Primary interface with the environment outside the computer system
- Deals with exchanging **blocks** of data
- Concerned with the placement of blocks
- Concerned with buffering blocks in main memory

Basic I/O Supervisor

- Responsible for all **file I/O** initiation and termination.
- Control structures deal with
  - Device I/O,
  - Scheduling,
  - File status.
- Selects and schedules I/O with the device

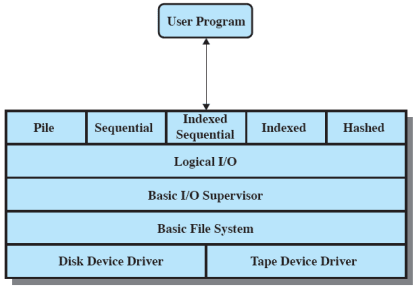


Figure 12.1 File System Software Architecture

Logical I/O

- Enables users and applications to access **records**
- Provides general-purpose record I/O capability
- Maintains basic data about file

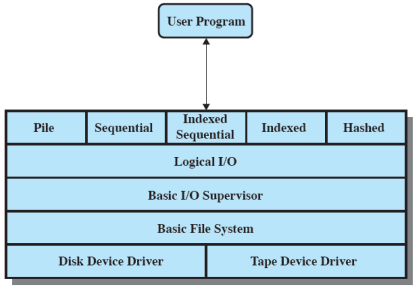


Figure 12.1 File System Software Architecture

Access Method

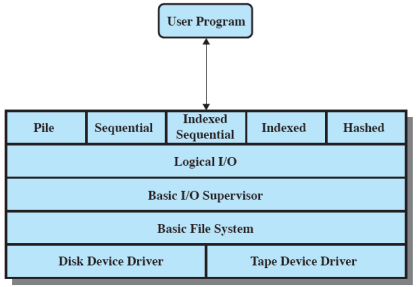
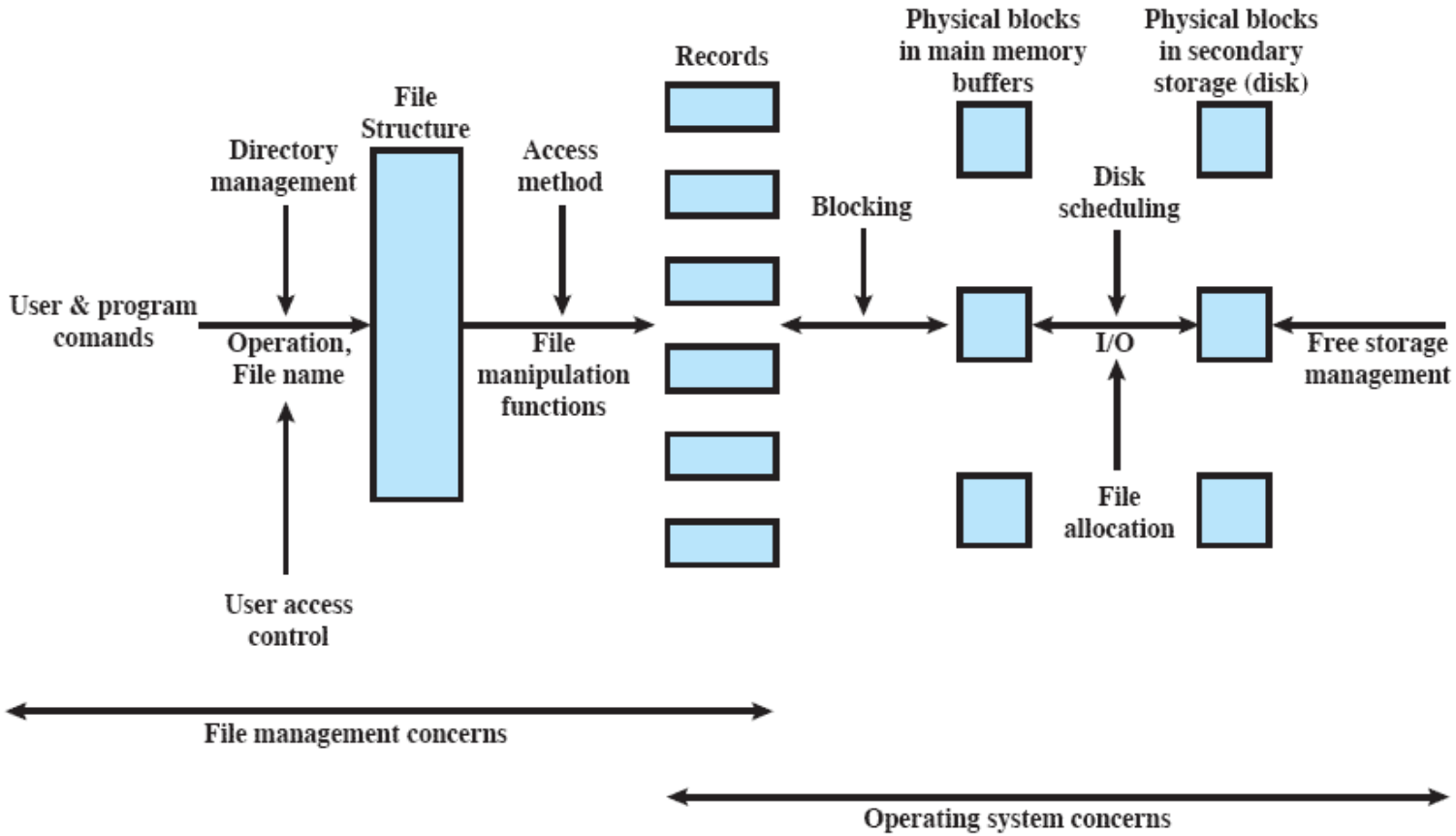


Figure 12.1 File System Software Architecture

- Closest to the user
- Provides a standard interface between applications and the file systems and devices that hold the data
- Access method varies depending on the ways to access and process data for the device.

Elements of File Management



Roadmap

- Overview
- File organisation and Access
- File Directories
- File Sharing
- Record Blocking
- Secondary Storage Management
- File System Security
- Unix File Management
- Linux Virtual File System
- Windows File System

Fontys

File Organization

- File Management Referring to the logical structure of records
  - Physical organization discussed later
- Determined by the ***way*** in which files are accessed

Criteria for File Organization

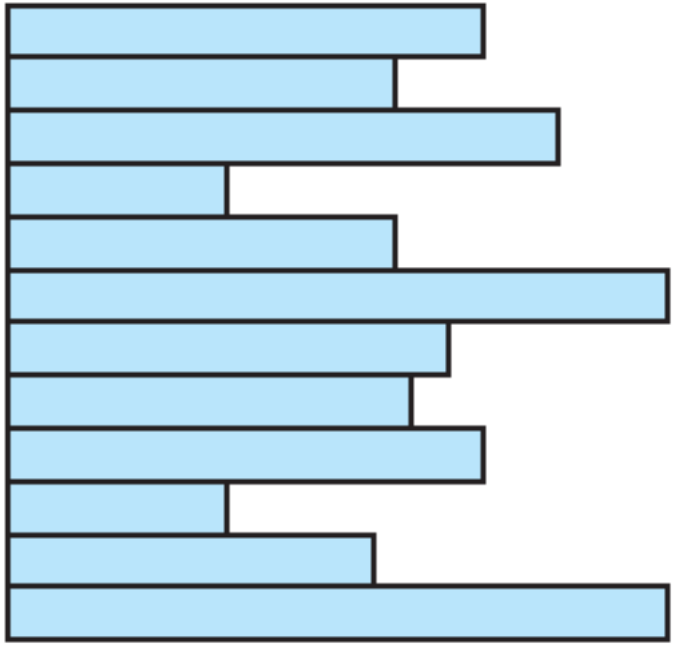
- Important criteria include:
  - Short access time
  - Ease of update
  - Economy of storage
  - Simple maintenance
  - Reliability
- Priority will differ depending on the use (e.g. read-only CD vs Hard Drive)
  - Some may even conflict

File Organisation Types

- Many exist, but usually variations of:
  - Pile
  - Sequential file
  - Indexed sequential file
  - Indexed file
  - Direct, or hashed, file

The Pile

- Data are collected in the order they arrive
  - No structure
- Purpose is to accumulate a mass of data and save it
- Records may have different fields
- Record access is by exhaustive search

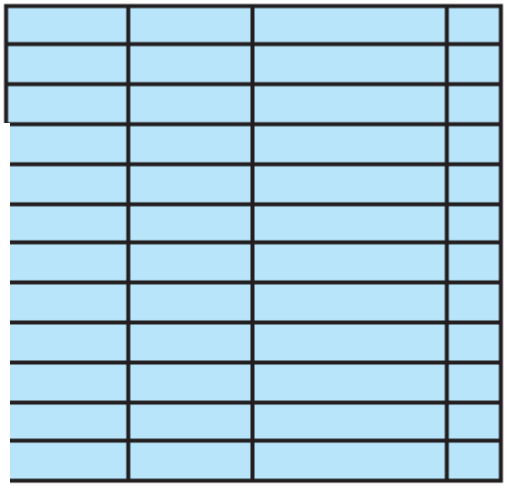


Variable-length records
Variable set of fields
Chronological order

(a) Pile File

The Sequential File

- Fixed format used for records
- Records are the same length
- All fields the same (order and length)
- Field names and lengths are attributes of the file
- Key field
  - Uniquely identifies the record
  - Records are stored in key sequence



Fixed-length records
Fixed set of fields in fixed order
Sequential order based on key field

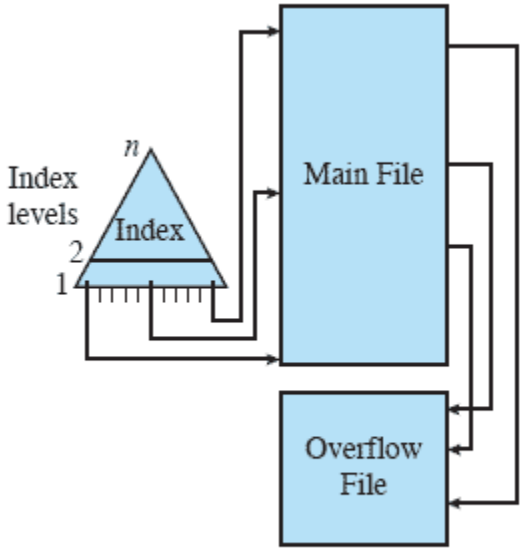
(b) Sequential File

Indexed Sequential File

- Maintains the key characteristic of the sequential file:
  - records are organized in sequence based on a key field.

Two features are added:

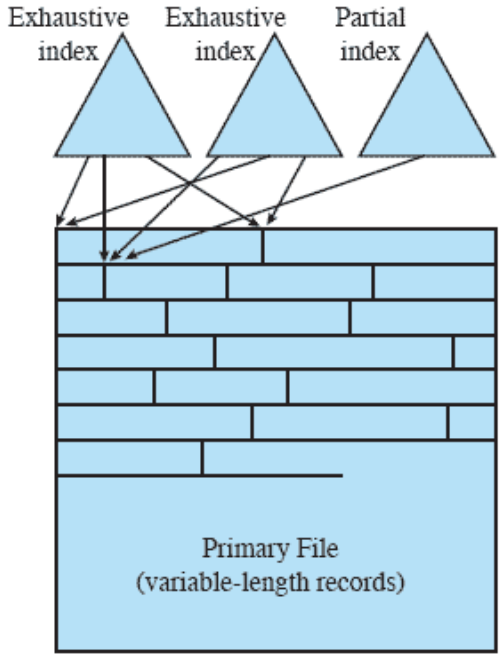
- an index to the file to support random access,
- and an overflow file.



(c) Indexed Sequential File

Indexed File

- Uses multiple indexes for different key fields
  - May contain an exhaustive index that contains one entry for every record in the main file
  - May contain a partial index
- When a new record is added to the main file, all of the index files must be updated.



(d) Indexed File

Roadmap

- Overview
- File organisation and Access
- File Directories
- File Sharing
- Record Blocking
- Secondary Storage Management
- File System Security
- Unix File Management
- Linux Virtual File System
- Windows File System

Fontys

File Directory

- Contains information about files
  - Attributes
  - Location
  - Ownership
- Directory itself is a file owned by the operating system
- Provides mapping between file names and the files themselves

Simple Structure for a Directory

- The method for storing the previous information varies widely between systems
- Simplest is a list of entries, one for each file
  - Sequential file with the name of the file serving as the key
  - Provides no help in organizing the files
  - Forces user to be careful not to use the same name for two different files

Two-Level Scheme for a Directory

- One directory for each user and a master directory
  - Master directory contains entry for each user
  - Provides address and access control information
- Each user directory is a simple list of files for that user
  - Does not provide structure for collections of files

Hierarchical, or Tree-Structured Directory

- Master directory with user directories underneath it
- Each user directory may have subdirectories and files as entries

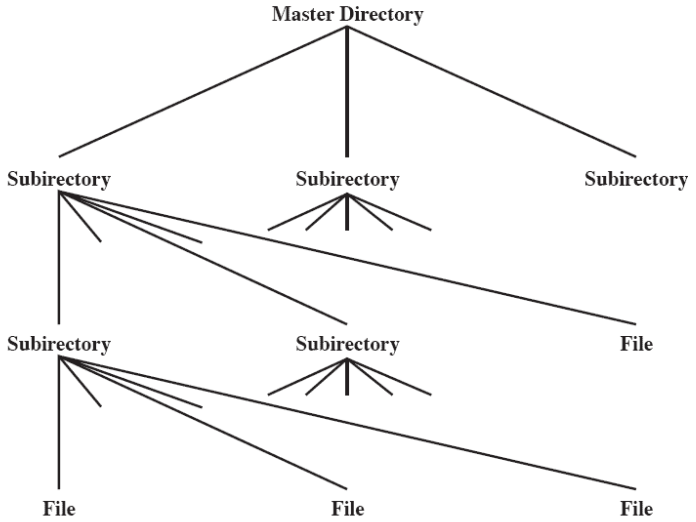


Figure 12.4 Tree-Structured Directory

Naming

- Users need to be able to refer to a file by name
  - Files need to be named uniquely, but users may not be aware of all filenames on a system
- The tree structure allows users to find a file by following the directory path
  - Duplicate filenames are possible if they have different pathnames

Example of Tree-Structured Directory

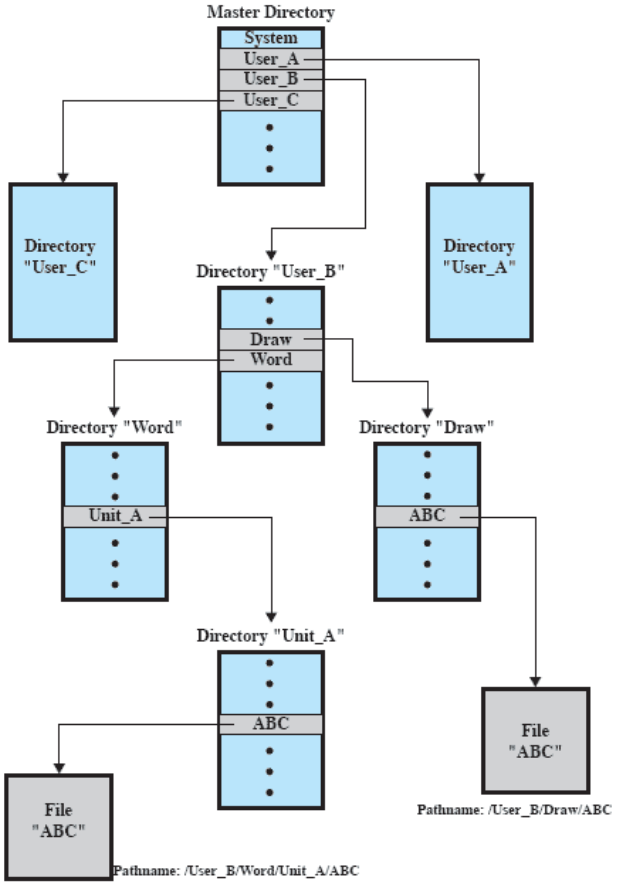


Figure 12.5 Example of Tree-Structured Directory

Working Directory

- Stating the full pathname and filename is awkward and tedious
- Usually an interactive user or process is associated with a **current** or **working directory**
  - All file names are referenced as being relative to the working directory unless an explicit full pathname is used

Roadmap

- Overview
- File organisation and Access
- File Directories
- File Sharing
- Record Blocking
- Secondary Storage Management
- File System Security
- Unix File Management
- Linux Virtual File System
- Windows File System

File Sharing

- In multiuser system, allow files to be shared among users
- Two issues
  - Access rights
  - Management of simultaneous access

Access Rights(1/3)

A wide variety of access rights have been used by various systems

- None
  - User may not even know of the files existence
- Knowledge of existence
  - User can only determine that the file exists and who its owner is

Access Right(2/3)

- Execution
  - The user can load and execute a program but cannot copy it
- Reading
  - The user can read the file for any purpose, including copying and execution
- Appending
  - The user can add data to the file but cannot modify or delete any of the file's contents

Access Right(3/3)

- Updating
  - The user can modify, delete, and add to the file's data.
- Changing protection
  - User can change access rights granted to other users
- Deletion
  - User can delete the file

User Classes

- Owner
  - Usually the files creator, usually has full rights
- Specific Users
  - Rights may be explicitly granted to specific users
- User Groups
  - A set of users identified as a group
- All
  - everyone

Simultaneous Access

- User may lock entire file when it is to be updated
- User may lock the individual records during the update
- Mutual exclusion and deadlock are issues for shared access

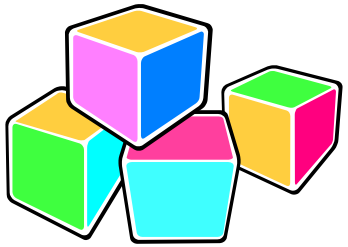
Roadmap

- Overview
- File organisation and Access
- File Directories
- File Sharing
- Record Blocking
- Secondary Storage Management
- File System Security
- Unix File Management
- Linux Virtual File System
- Windows File System

Blocks and records

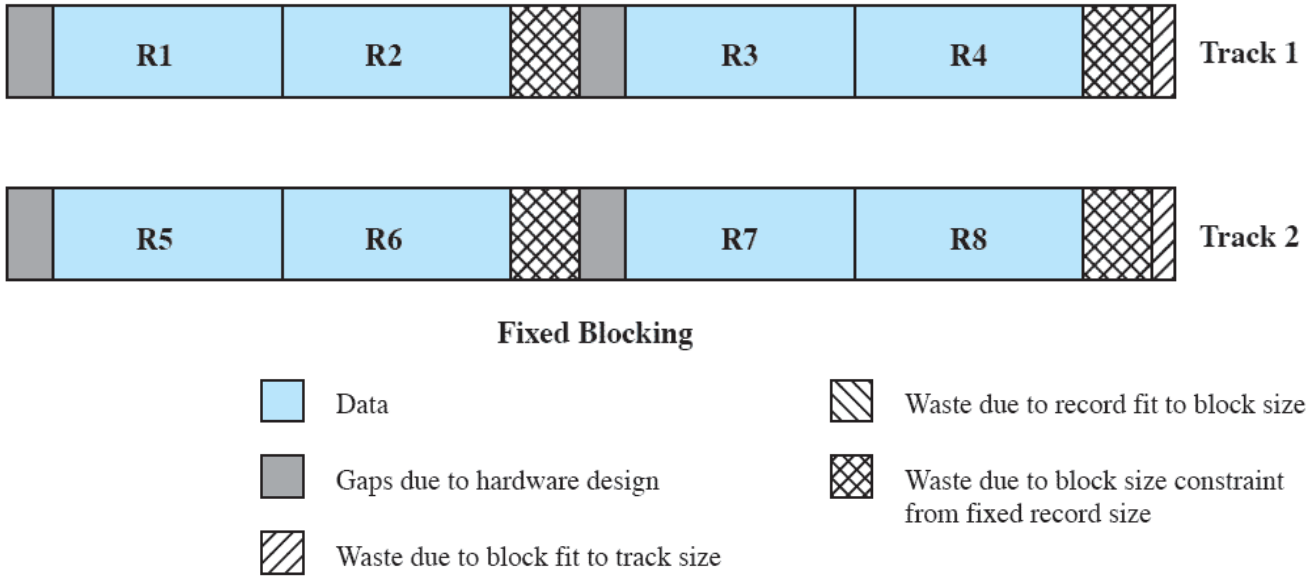
- Records are the logical unit of access of a structured file
  - But blocks are the unit for I/O with secondary storage
- Problem: how to pack records into blocks?
- Three approaches are common
  - Fixed length blocking
  - Variable length spanned blocking
  - Variable-length unspanned blocking

  Note: “blocking“ in the sense of “packing“



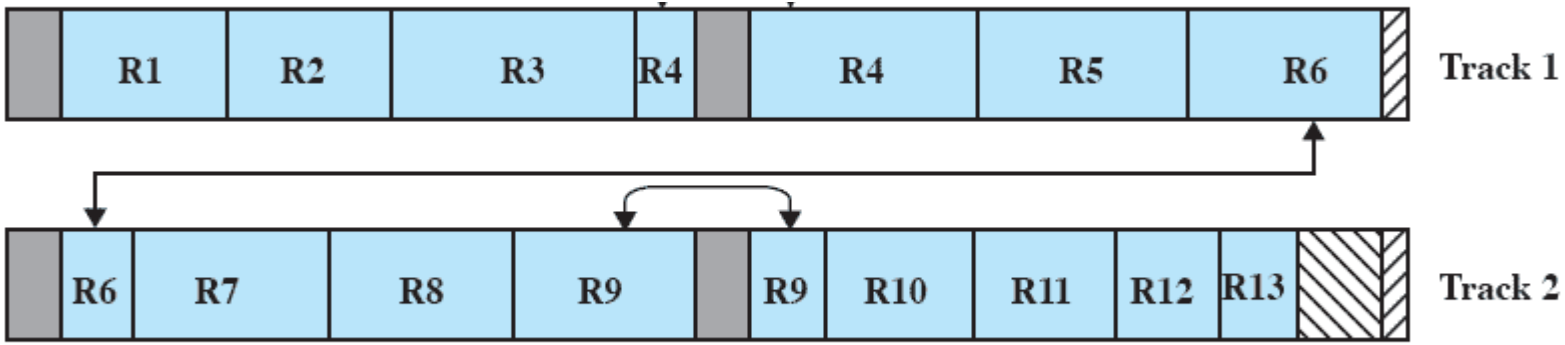
Fixed Blocking

- Fixed-length records are used, and an integral number of records are stored in a block.
- Unused space at the end of a block is internal fragmentation



Variable Blocking: Spanned

- Variable-length records are used and are packed into blocks with no unused space.
- Some records may span multiple blocks
  - Continuation is indicated by a pointer to the successor block



Variable Blocking: Spanned

Data

Gaps due to hardware design

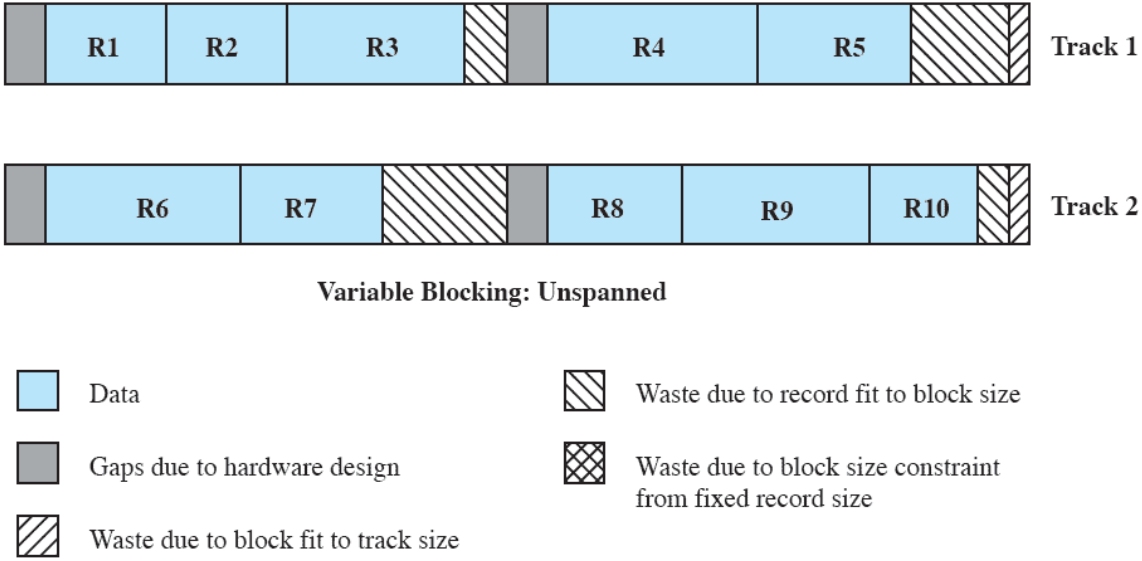
Waste due to block fit to track size

Waste due to record fit to block size

Waste due to block size constraint from fixed record size

Variable Blocking: Unspanned

- Uses variable length records without spanning
- Wasted space in most blocks because of the inability to use the remainder of a block if the next record is larger than the remaining unused space.



Roadmap

- Overview
- File organisation and Access
- File Directories
- File Sharing
- Record Blocking
- Secondary Storage Management
- File System Security
- Unix File Management
- Linux Virtual File System
- Windows File System

Fontys

Preallocation (vs Dynamic Allocation)

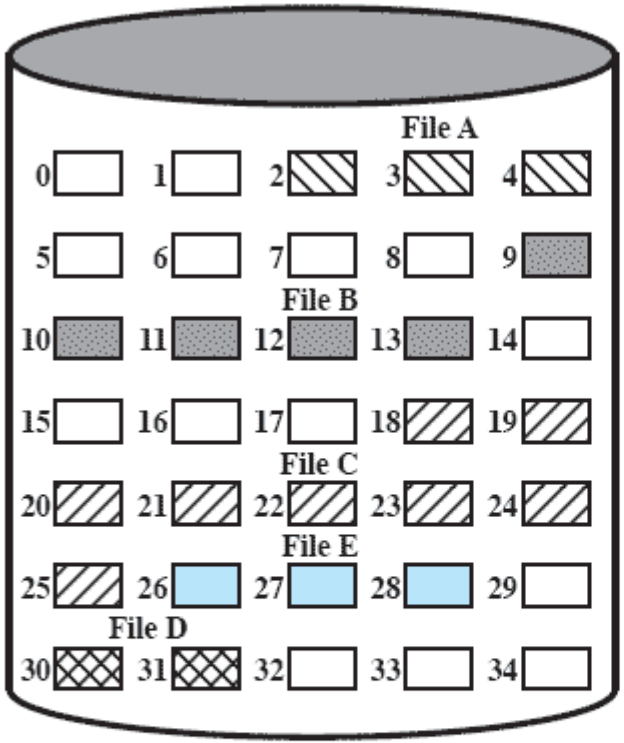
- Need the maximum size for the file at the time of creation
- Difficult to reliably estimate the maximum potential size of the file
- Tend to overestimated file size so as not to run out of space

So: …..

File Allocation Method

- Three methods:
  - contiguous,
  - chained, and
  - indexed.

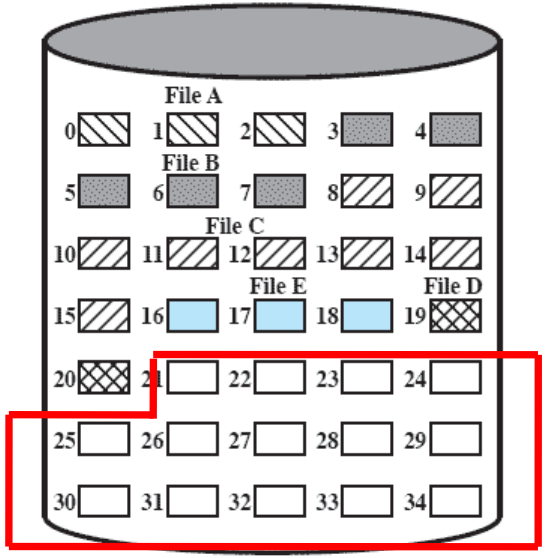
Contiguous File Allocation



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

After compaction



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)

Chained Allocation

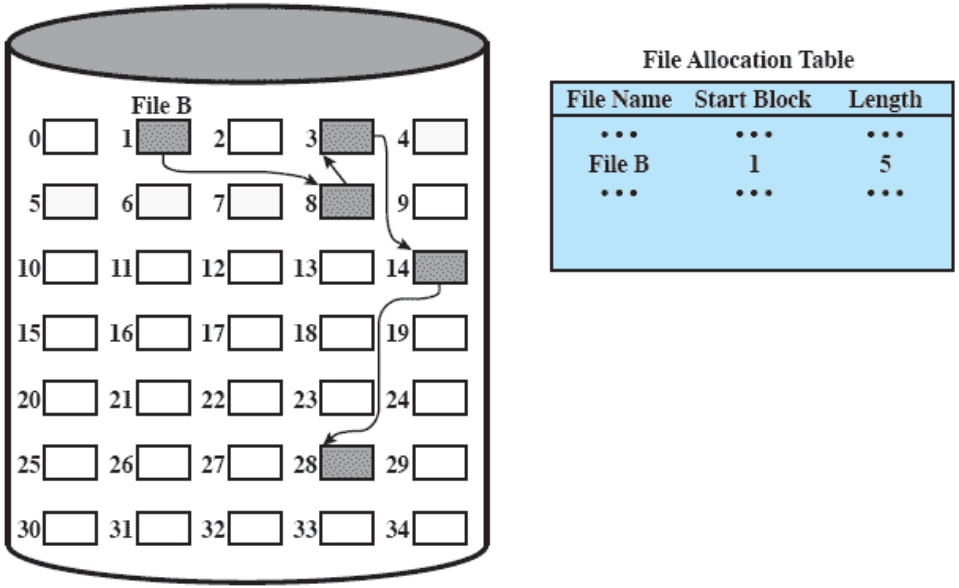
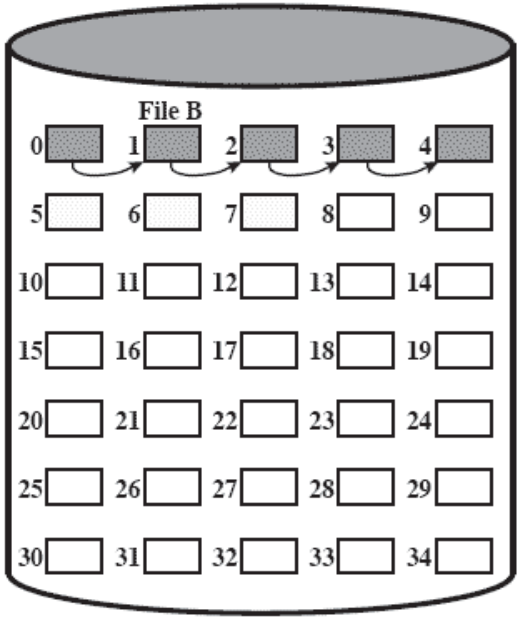


Figure 12.9 Chained Allocation

Chained Allocation Consolidation



File Allocation Table

File Name	Start Block	Length
• • •	• • •	• • •
File B	0	5
• • •	• • •	• • •

Figure 12.10 Chained Allocation (After Consolidation)

Indexed Allocation

- File Allocation Table (FAT) 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

Indexed allocation with Block Portions

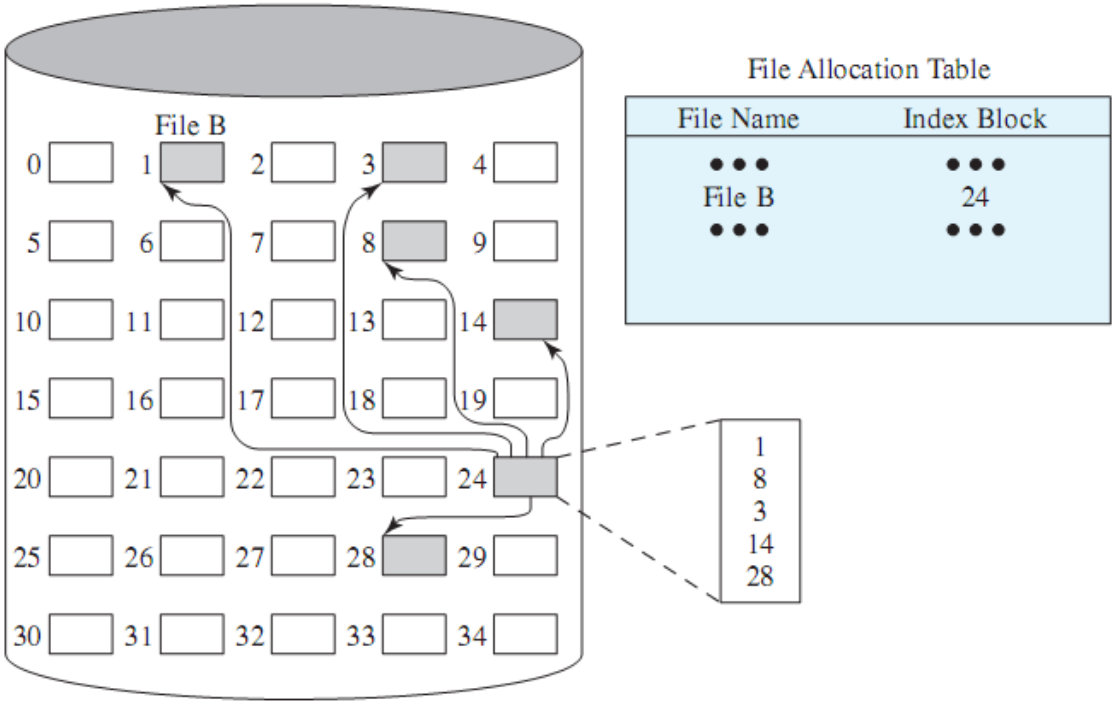


Figure 12.11 **Indexed Allocation with Block Portions**

Indexed Allocation with Variable Length Portions

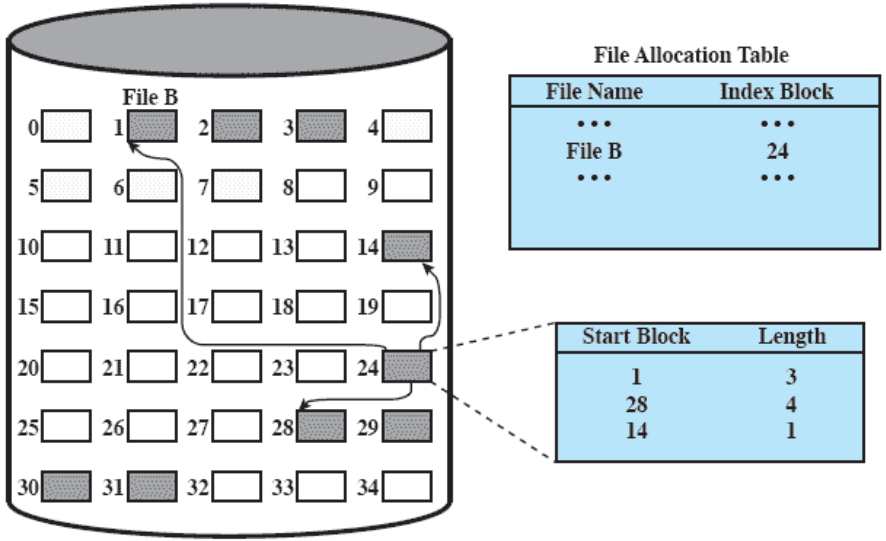


Figure 12.12 Indexed Allocation with Variable-Length Portions

Roadmap

- Overview
- File organisation and Access
- File Directories
- File Sharing
- Record Blocking
- Secondary Storage Management
- File System Security
- Unix File Management
- Linux Virtual File System
- Windows File System

UNIX File Management

- Six types of files

Regular, or ordinary
- contains arbitrary data in zero or more data blocks

Directory
- contains a list of file names plus pointers to associated inodes

Special
- contains no data but provides a mechanism to map physical devices to file names

Named pipes
- an interprocess communications facility

Links
- an alternative file name for an existing file

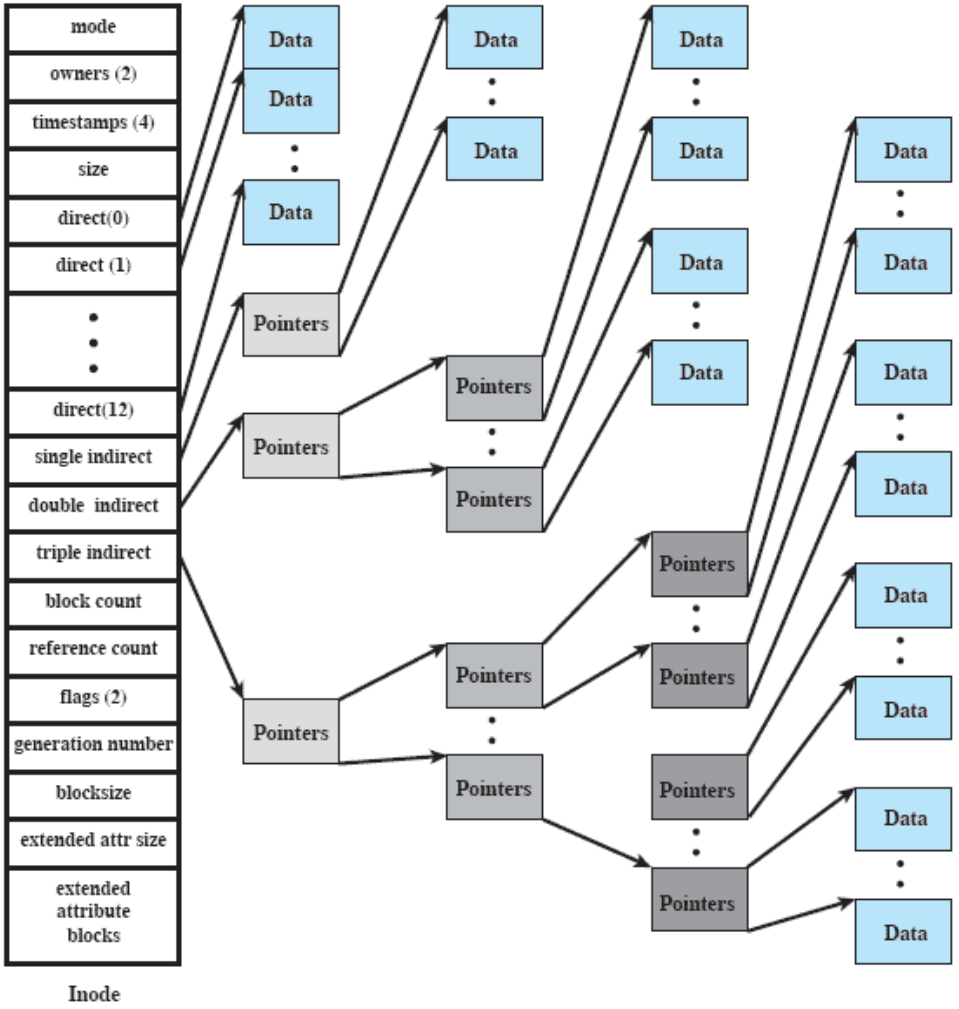
Symbolic links
- a data file that contains the name of the file it is linked to

INodes

- All types of UNIX files are administered by the OS by means of inodes
- An inode (index node) is a control structure that contains the key information needed by the operating system for a particular file
- Several file names may be associated with a single inode
    - an active inode is associated with exactly one file
    - each file is controlled by exactly one inode

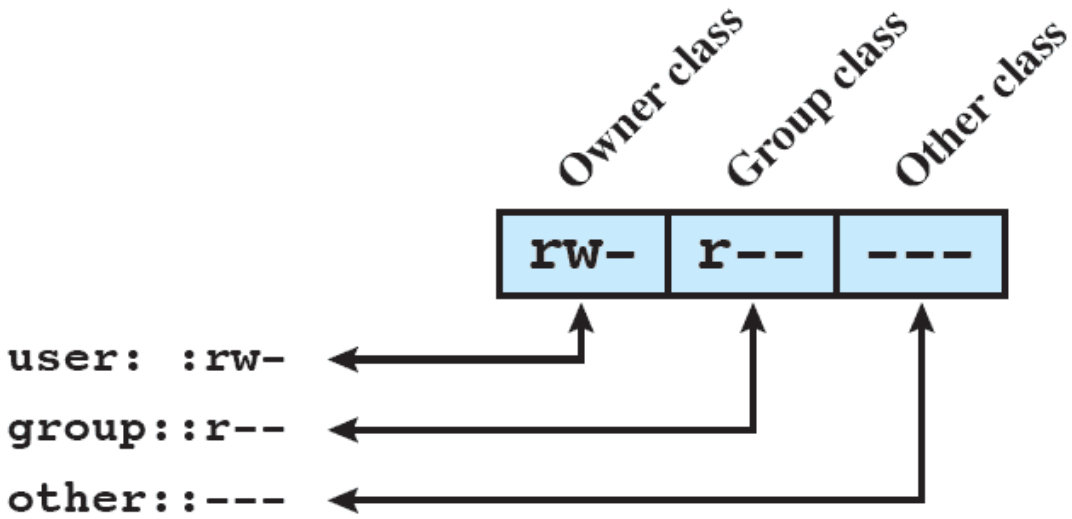
Fontys

FreeBSD Inode and File Structure



Fontys

Unix File Access Control



(a) Traditional UNIX approach (minimal access control list)

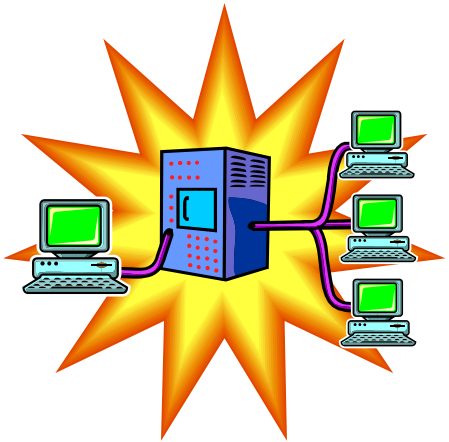
Fontys

Roadmap

- Overview
- File organisation and Access
- File Directories
- File Sharing
- Record Blocking
- Secondary Storage Management
- File System Security
- Unix File Management
- Linux Virtual File System
- Windows File System

Windows File System

- Key features of NTFS
  - Recoverability
  - Security
  - Large disks and large files
  - Multiple data streams
  - Journaling
  - Compression and Encryption
  - Hard and symbolic links



NTFS Volume and File structure

- NTFS makes use of the following disk storage concepts

Sector

- the smallest physical storage unit on the disk
- the data size in bytes is a power of 2 and is almost always 512 bytes

Cluster

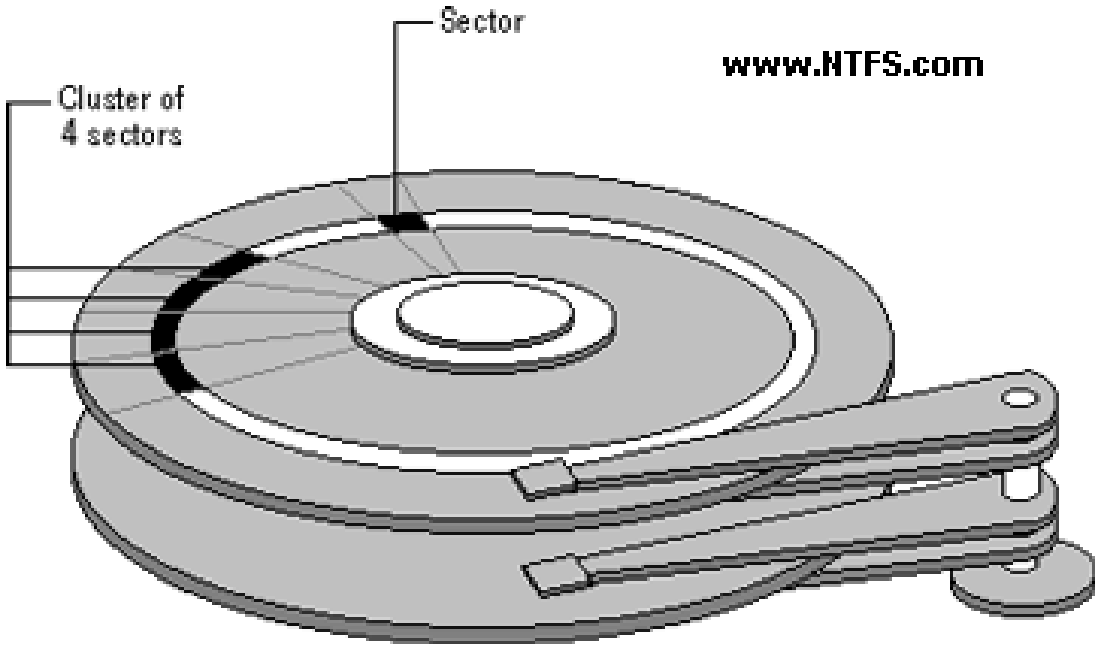
- one or more contiguous sectors
- the cluster size in sectors is a power of 2

Volume

- a logical partition on a disk, consisting of one or more clusters and used by a file system to allocate space
- can be all or a portion of a single disk or it can extend across multiple disks
- the maximum volume size for NTFS is 264 bytes

Fontys

Sector – Cluster (Volume)



Fontys

Windows NTFS Components

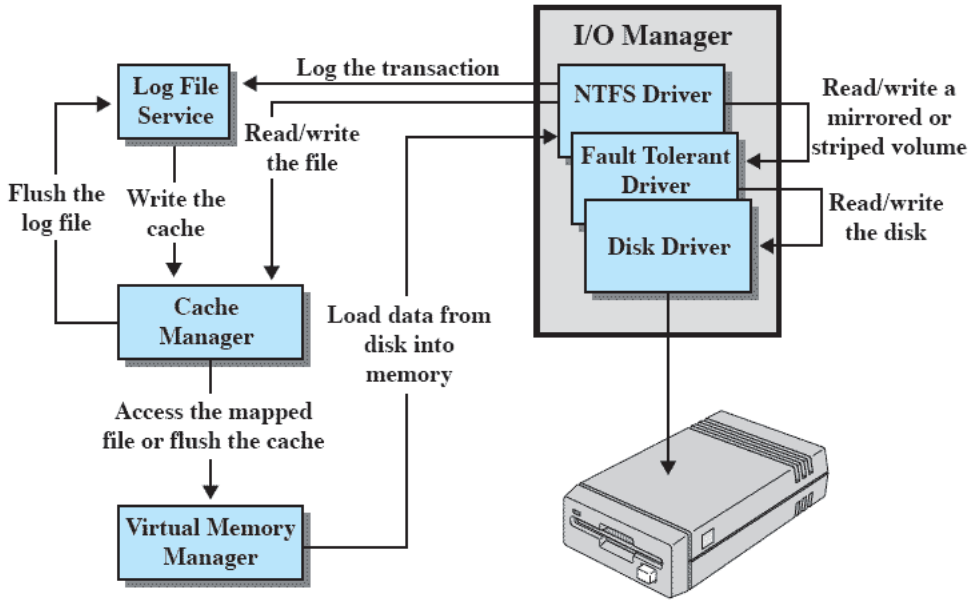
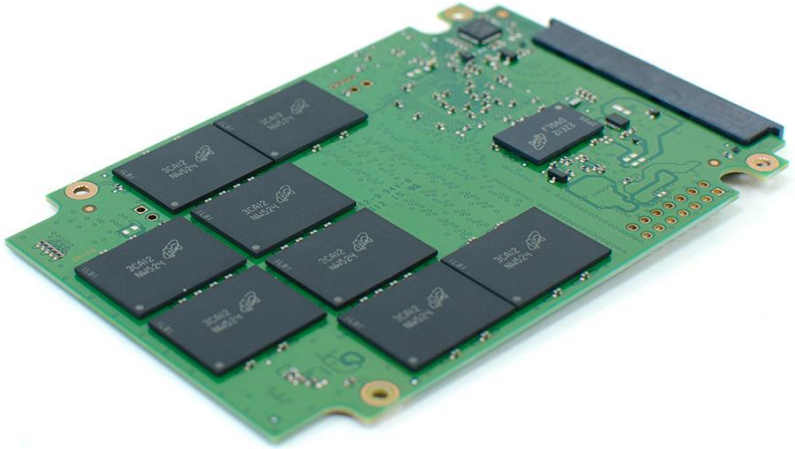


Figure 12.20 Windows NTFS Components

Fontys

› FOR SOCIETY

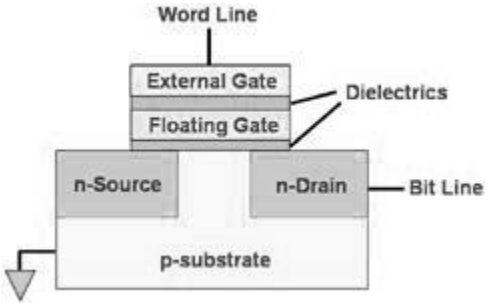
SSD



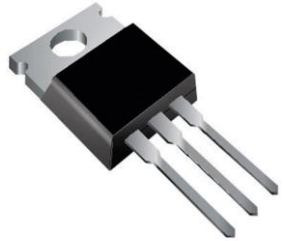
SSD – Solid State Drive

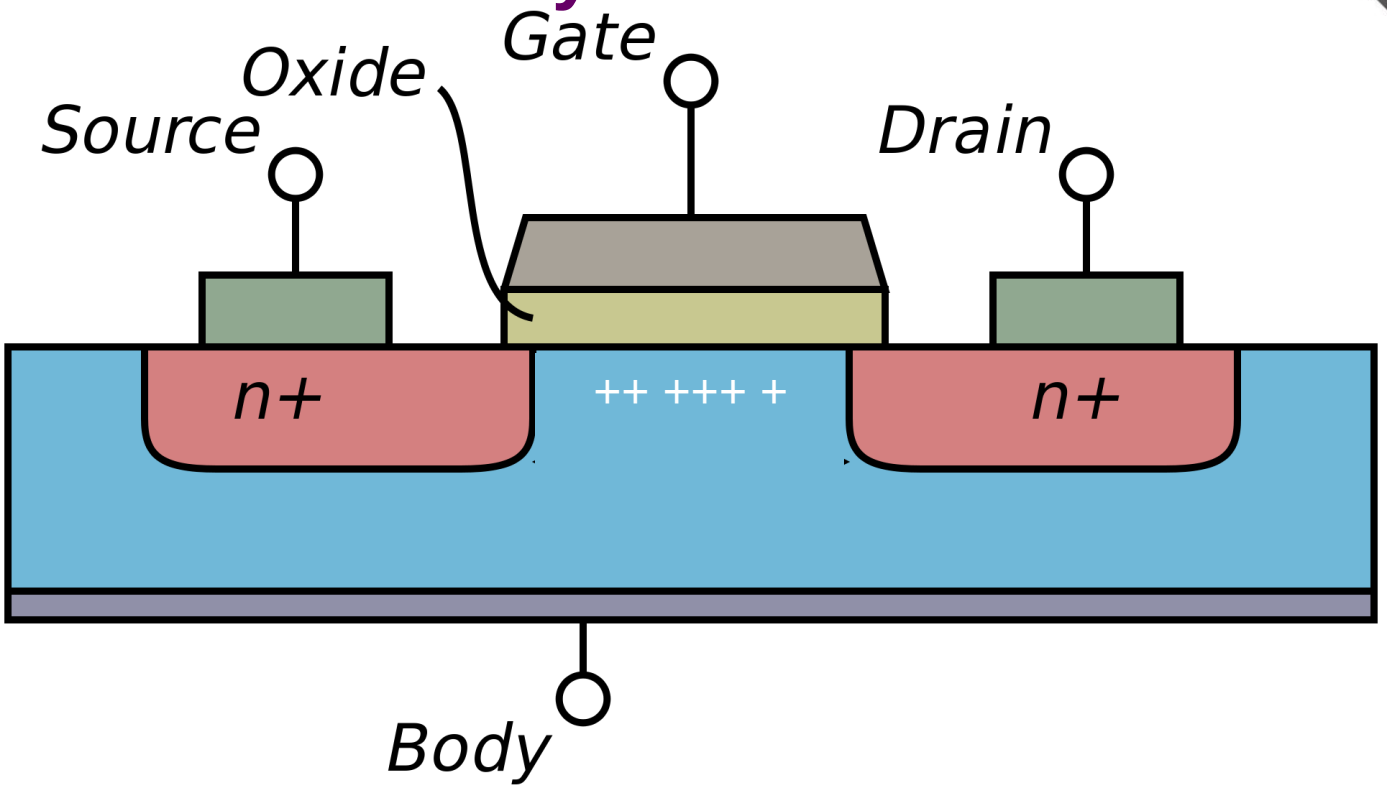
- NAND Flash memory
  - Non volatile - Keeps information when power switched off
  - No defragmentation needed
- Lower Power consumption
- No moving parts

- Memory wears out
  - Ordinary server/workstation use: >10 years
  - Only for high-end server systems
  - MTBF for SSD > HDD

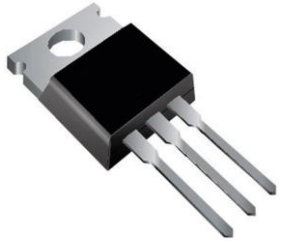


Why wear – Ordinary MOSFET

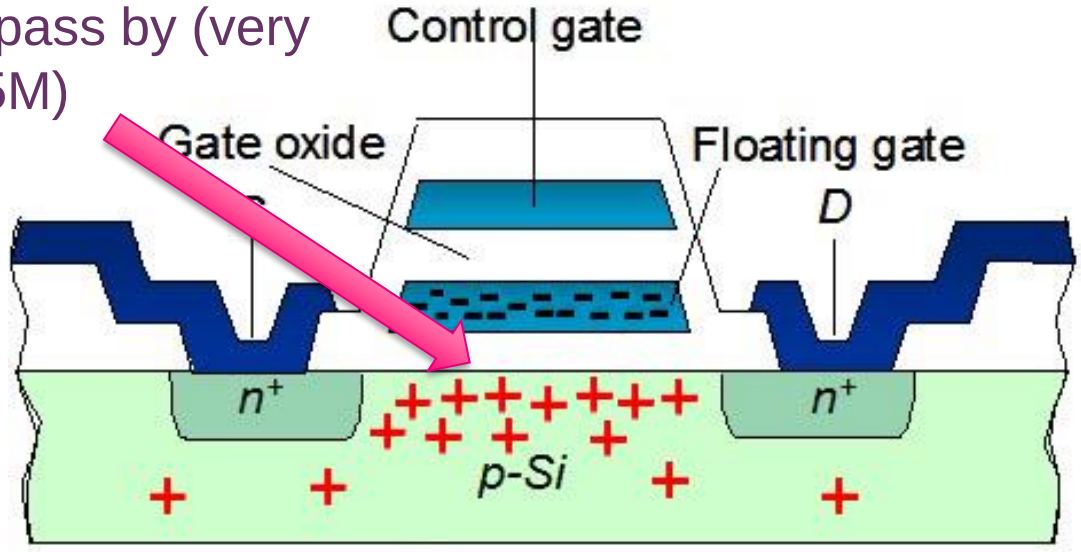




Why wear – Floating Gate MOSFET



Insulator becomes conductive after electrons pass by (very many times (>5M)



Comparison

Device	Example	Time to access	Cost per bit
L1 Cache	i7	1,2 ns	?
DRAM	Kingston 4GB - DDR3	13,75 ns	1,16 n€
SSD	Samsung 500GB	~10.000 ns	0,075 n€
HDD Disk Drive	Seagate Desktop HDD 4TB Sata	~5.000.000 ns	0,0046 n€