File system — logical layer

Definition of file

- A file is a named collection of related information that is recorded in secondary storage. It is an abstract representation of data stored in a computer system.
- The task of an OS it to map the abstract view of the file and concrete data representation on storage.

File attributes

- name — to identify a file by human users
- type — to understand/interpret the contents
- location — to find the contents in the storage
- size — in particular units (bytes, words, blocks)
- protection — to control access rights
- date/time — the date and time of specific operations (e.g. creation, last modification, last access)

File type

- The file type determines the kind of information stored in the file, in particular the interpretation/understanding of its contents.
- Information of type can be located in the internal structures of the file system, within the contents of the file itself, in the directory entry as one of its fields, or the file name (the file name is usually also a field in the directory entry).
- The file type may be recognised (possibly supported) by the operating system; otherwise, it is only information for the user or application.

File structure

- Logical file structure
    - indicated by the file type
    - specifies the internal organisation of data inside the file;
    - may be recognised and understood by:
        - the operating system kernel, or
        - an application (from the OS kernel view point it is only a set of bytes).
    - It defines the structure of the abstract view of the file, accessible to the programmer.
    - Logical record — a single unit of information in the abstract logical file structure. In UNIX it is for example 1 byte.
- Physical file structure
    - specifies the organisation of data on a storage (usually a set of blocks);
    - imposed by a storage device.
    - OS usually maps the logical file structure into the physical file structure by packing many logical records into a single physical block. Physical block is a block in the secondary memory, it is a unit of information in the physical file structure.
    - Internal fragmentation — because the OS packs many single logical records into a few big physical blocks, the last physical block is usually not full. This waste is called internal fragmentation.

Access methods

- The access method is the way the data is read from or written to a file.
- The information in the file can be accessed in several ways.
- Support for a given access method is derived from the properties of a storage device or file structure (physical or logical).
- Two methods are usually distinguished:
    - Sequential access:

The information in the file is processed in order, one record after the other. Read or write to a file automatically advances the file pointer which tracks the location in the file. This is the most common method.

- Direct access:

  In this model we treat the file as a sequence of fixed length logical records. The direct access method allows us to read or write any segment of the file without a particular order. For example we may read block 14, then read block 53, then write block 7. This method is of great use for accessing a large amount of information like in a database.

Common file access operations

Operation	Description	UNIX-like interface
Creating a file	Creating a file and corresponding directory entry	`creat` or `open`
Reading a file	Reading a fragment of file contents	`read`, `pread`
Writing a file	Writing (overwriting or appending) some data	`write`, `pwrite`
Truncation a file	Erasing a tail part of a file	`truncate` or `ftruncate`
Deleting a file	Erasing the directory entry and file contents (realising the blocks)	`unlink`

- Additional file operations:
  - `open` — opening a file to get the file descriptor
  - `close` — complementary to open
  - `lseek` — change the current offset (position pointer)

File system organisation

Partition

- Before an OS can use a drive to hold files it needs to record its own data structures on the device.

- OS partitions the disk in one or more groups of blocks or pages. Those partitions are treater as they were a separate device. For instance, one partition can hold a file system containing a copy of the operating system’s executable code and another a file system containing the user files.
- The partition information is written in a fixed format at a fixed location on the storage device.
- Partitioning allows the use of different filesystems to be installed for different kinds of files. Separating user data from system data can prevent the system partition from becoming full and rendering the system unusable.

Volume

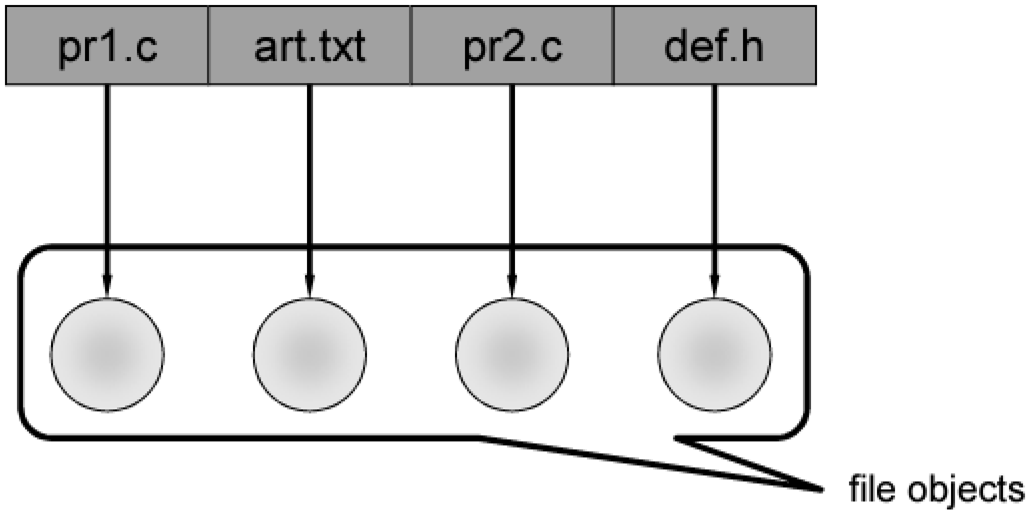
- After creating the partitions, the OS creates volumes.
- Volume is a single accessible storage with a single file system.
- A volume typically resides on a single partition, although it can span multiple partitions or disks. This differs a volume and a partition, since a partition can’t combine multiple disks.
- In simple terms a volume is an abstract storage unit in the file system, while a partition is a physical low-level division of a disk.

Directories

- The directory can be viewed as a symbol table that translates file names into their file control blocks.
- A directory can be organized in many ways

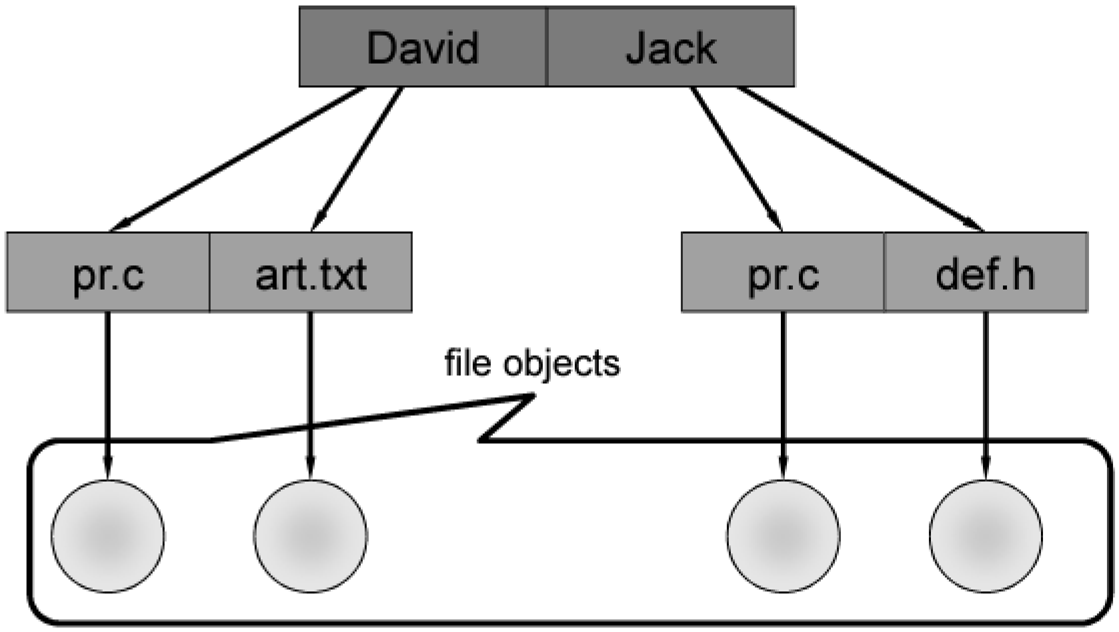
Single-level directory

- This is the simples directory structure.
- All files are contained in the same directory. Even if there are multiple users, all their files are contained in the same directory.
- This organization has many limitations, since all files must have unique names.



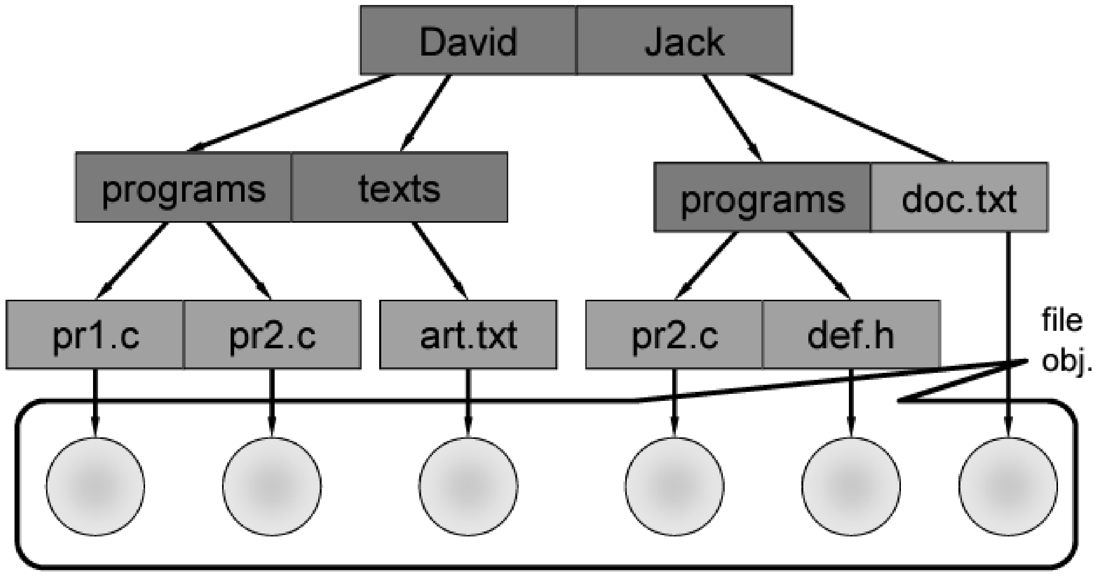
Two-level directory

- In two-level directory structure, each user has his own directory which lists only his own files.
- When a user logs in, their own directory is indexed and when the user searches for a file, only this directory is searched.
- The disadvantage here is that it isolates one user from another, since this organization doesn't allow a user to access another user's data.



Tree-structured directory

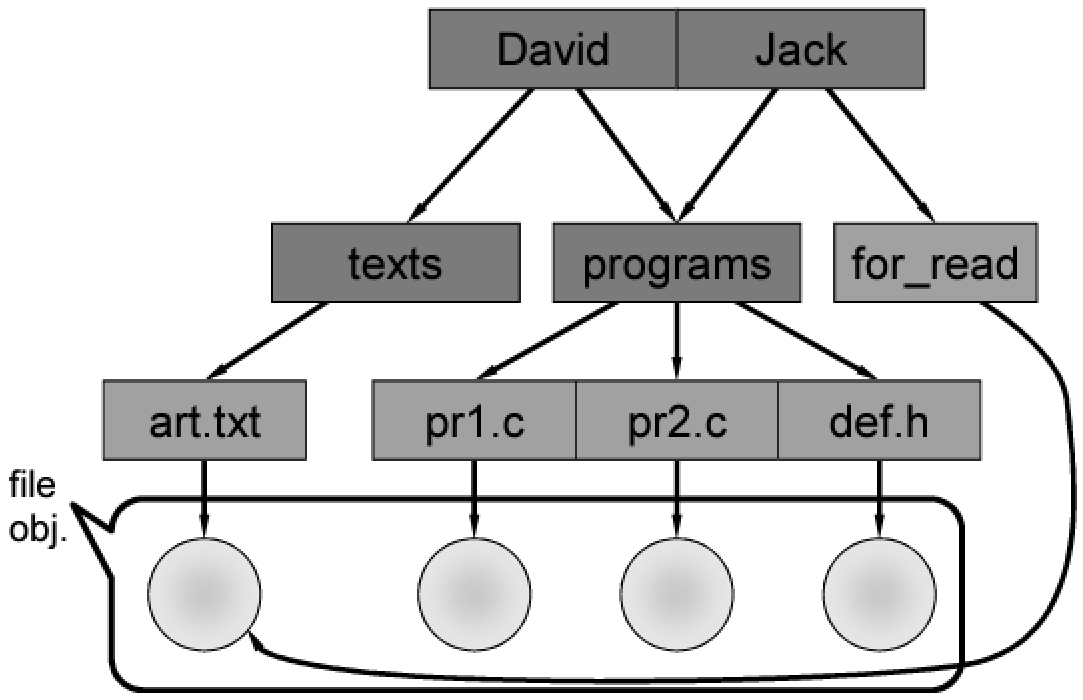
- In tree-structured directory organization a user can create their own subdirectories and organize their files accordingly.
- This is the most common directory structure.
- This tree has a root directory and every file in the system has a unique path name.
- In many implementations a directory is simply another file, that is treated in a special way.



Acyclic-Graph directories

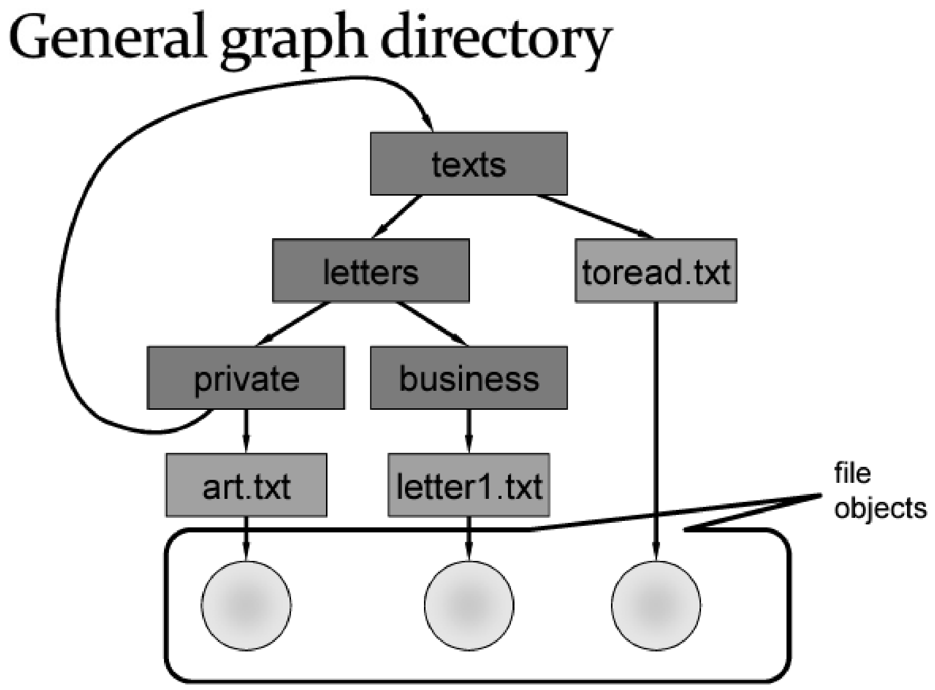
- This organization allows to users to share a subdirectory or a file.
- The same file may be in two different directories.
- Even a single user can place one file into two different subdirectories.
- This organization is implemented by having a special file called a **link** which stores a pointer to another file or subdirectory, like for example a path name.

Acyclic-graph directory



General graph directory

- A problem with acyclic-graph directory is ensuring that there are no cycles.
- The general graph directory allows cycles to exist.



Directory operations

- Directory creation — a special case of file creation, as a directory is a special purpose file (of a particular type).
- Directory deletion — allowed for empty directories or recursively deleting all directory contents.
- Entry creation — rather a side effect of other operations (e.g. file creation) than stand-alone one.
- Search for entry — an operation to find an entry of a given name.
- Renaming —change the name of an entry.
- Listing — obtaining the list of entries.