

```
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
```

```
#define MAXPAROLA 30
#define MAXRIGA 80
```

```
int main(int argc, char *argv[])
```

```
{
    int freq[MAXPAROLA]; /* vettore di contatori
delle frequenze delle lunghezze delle parole */
    char riga[MAXRIGA];
    int i, inizio, lunghezza;
    FILE * f;
```

```
    for(i=0; i<MAXPAROLA; i++)
        freq[i]=0;
```

```
    if(argc != 2)
```

```
    {
        fprintf(stderr, "ERRORE: serve un parametro con il nome del file\n");
        exit(1);
    }
```

```
    f = fopen(argv[1], "rt");
    if(f==NULL)
```

```
    {
        fprintf(stderr, "ERRORE: impossibile aprire il file %s\n", argv[1]);
        exit(1);
    }
```

```
    while( fgets( riga, MAXRIGA, f ) != NULL )
```

## File-System

### Directories in Linux

Stefano Quer - Pietro Laface

Dipartimento di Automatica e Informatica

Politecnico di Torino

## Directories

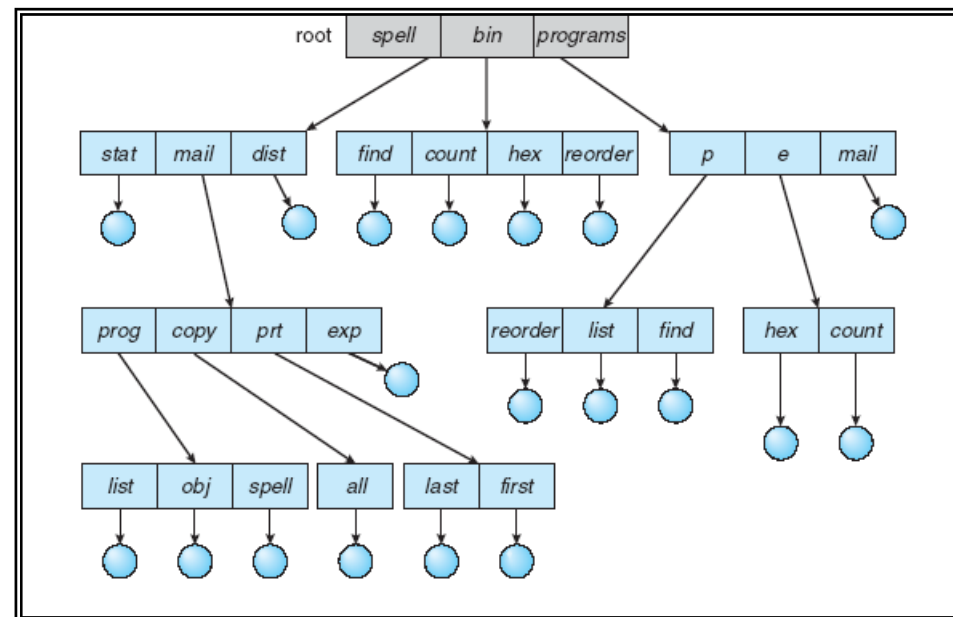
- ❖ Modern file systems are structured by means of special files: the directories
  - A directory is the node of a tree, or the vertex of a graph that includes some information about the (regular) files that it contains.
  - Directories and files are stored in the mass memory
- ❖ Operations that can be performed on directories are similar to the ones applied to files
  - Creation, deletion, listing, rename, visit, search, etc.

## Structure

- ❖ Structuring a file systems by means of directories has several advantages:
  - Efficiency
    - Searching a file
  - Naming
    - Simplicity for a user to identify his files
      - The same name can be assigned to different files
  - Grouping (organization)
    - Grouping to the programs and data according to their characteristics
      - Editors, compilers, documents, etc.

## Directory tree

- ❖ Directories and files are organized as a tree
  - Every node/vertex of the tree can include other nodes/vertex of the tree



## Tree directories

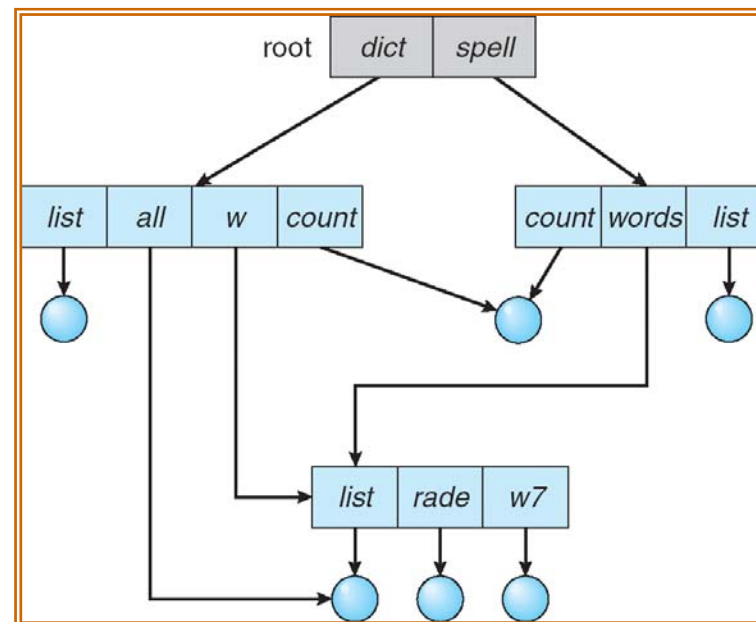
- ❖ Tree directories imply:
  - Current work directory, change of directory, absolute and relative path name, etc.
- ❖ Performance
  - Efficiency
    - Efficient search
  - Naming
    - With absolute path or relative to the current working directory
  - Grouping
    - Flexible

## Acyclic graph directories

- ❖ A tree file system does not allow **sharing**
- ❖ It is often useful to refer to the same object in the file system with different filenames
  - Same user refers to an object with different pathnames
  - Different users want to share objects
  - It is worth noting that duplication of the object is not a solution because of
    - Increase of file system occupation
    - Possible information incoherence in one or more copies

## Acyclic graph directories

- ❖ Tree file systems can be generalized organizing them as acyclic graphs.



## Acyclic graph directories

### ❖ Performance

- Similar to tree directories, but with possibility of sharing

### ❖ Method

- A directory entry can be share by means of the creation of la **link**
  - A link is a reference (pointer) to another pre-existing entry



## Acyclic graph directories

### ❖ More complex filesystem management due to links

#### ➤ File system visit

- If the entry is a link, follow the indirect link to reach the pointed-to entry
- Many absolute paths, and different filenames may correspond to the same entry
  - File system analysis (statistics, e.g., how many files with extension ".c" exist?) are more complex

## Acyclic graph directories

### ➤ Removing an entry)

- Immediate deletion of the data
  - Some dangling links may remain
  - Attempt accessing data through a link could return a missing file error
- Remove the data when the last link is removed
  - Avoids dangling links
  - Need to manage multiple links
    - Keeping the list of all link is expensive (variable length list)
    - Removing all links is expensive (need to search)
    - Search is more expensive
  - It is more effective to store a link counter associated to the data (for the hard links in UNIX the counter is stored in a field of the inode)

## Acyclic graph directories

- ❖ Creating a new link to a directory could cause the generation of a cycle in the file system
  - Managing a cyclic graph is more complex
    - Search and visit has to avoid infinite recursion
  - The simplest strategy avoid visiting the links
  - Cycles can be avoided if the creation of a link to a directory is forbidden