Tlemcen University
Faculty of Science
Computer Science Department
1st Year Engineer

Academic year: 2023-2024
Introduction to operating systems 1

TP n°3: Physical and symbolic links and i-nodes

1. Objectives

The aim of this practical course is to explore in a little more detail the possibilities of the UNIX file system through the concept of links (physical links and symbolic links) and the notion of node index (inode) as well as to find out how certain commands are processed.

2. Physical and symbolic links

Links are useful for making the same file appear in several directories, or under different names. They help to avoid duplication (content appearing in several places) and ensure that updates are consistent, i.e. any update operation on the content of the source file will be shown in the other files. There are two types of links:

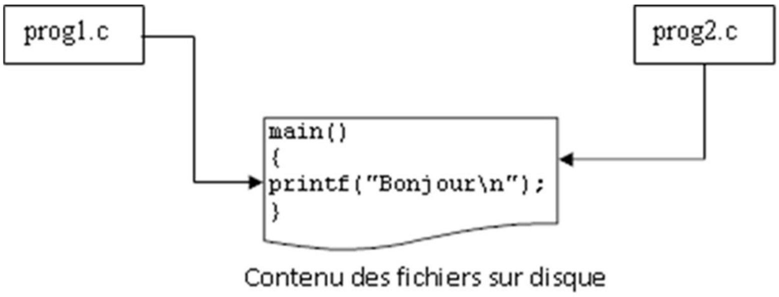
2.1 Physical links

This type of link allows two or more files to be associated with the same space on the disk. This operation is performed using the command ***ln***.

Example:

```
ln  prog1.c /home/student/prog2.c
```

The file ***"prog2.c"*** is created in the directory ***/home/student***. You can see that these 2 files are the same size. In terms of management, they are independent, but share the same disk space. Any modification to one modifies the other! But removing one breaks the link, but does not physically remove the other. Figure 1 shows how the two files appear and how they point to the same disk space.



```
main()
{
printf("Bonjour\n");
}
```

Figure 1: ***Illustration of physical links***

2.2 Symbolic links

The symbolic link refers to a file and removing the source file will change the behavior of the link file, which will be considered 'broken'. This operation is performed using the command ***ln -s***.

Example:

```
ln -s prog1.c /home/student/prog2.c
```

The ***ls -F*** command passed in the directory ***/home/student*** shows that the file prog2.c points to prog1.c (so a request to prog2.c will open prog2.c). Figure 2 shows how the two files (source file and symbolic link) appear.

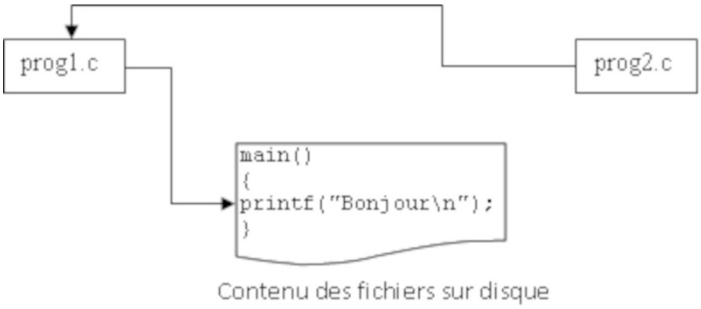


Figure 2: Illustration of symbolic links

3. i-nodes

Each file has a block of information associated with it, called an i-node (index node or i-node) containing a certain amount of general information about the file:

- its size,
- the address of the blocks used on the disk for this file,
- the identification of its owner and access permissions for different users,
- its type (ordinary or special file, directory etc.),
- a counter of its references in the system,
- a number of dates relating to the main operations that can be performed on the file (consultation, modification, etc.),
- and pointers to data blocks.

This information block does not contain a name for the file.

On a UNIX system, a directory is considered as a file. The latter is also identified by an inode, containing a list of inodes wherein each one represents a file.

To find out the inode number of a file, simply type the following command:

```
ls -i my-file
```

For example, in the ext2 file system, each inode contains around 64 fields, 13 of which contain blocks of two types:

- Address blocks, which contain pointers to other blocks;

- Data blocks, which contain the file data.

The first 10 fields (of the 13) contain the addresses of the first 10 blocks of data in the file (one address per block). If the blocks to which the first 10 fields point are sufficient to contain the file, fields 11, 12 and 13 are not used. Otherwise, in addition to the first 10 blocks, blocks 11, 12 and 13 are used. These blocks operate according to a system of indirection. There are three levels of indirection:

- simple indirection, used by field 11,
- double indirection, used by field 12,

- triple indirection, used by field 13.

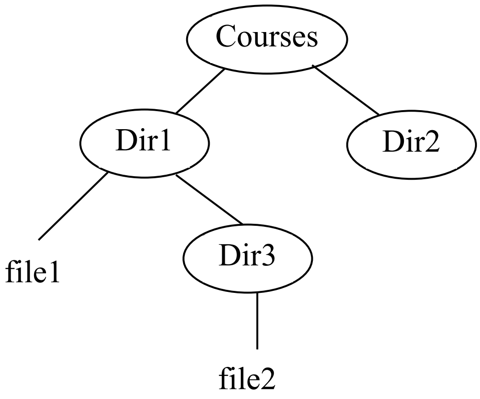
The difference between a physical link and a symbolic link lies in the inode: a physical link does not have its own inode, it has the inode of the file to which it points. A symbolic link, on the other hand, has its own inode.

It is not possible to create physical links between two different disk partitions, but with symbolic links this constraint does not apply.

Exercises

Exercise n° 1 : Creating links on files

1) Create the following tree structure in your working directory.



2) Using the command ***echo***, write "Third TP on links and i-nodes" in the file ***file2***.
3) Create a physical link to the file ***file2*** in the directory ***Courses*** under the name ***fiile2-phy***.
4) List the contents of the directory ***Dir1*** with details of each item.
5) Modify the contents of the file ***file2-phy*** using the command ***echo***. What do you notice for the file ***file2***? In a similar way, modify the file ***file2*** and consult the file ***file2-phy***. What do you notice?
6) Change the access permissions for the file ***file2*** for the members of the group. What do you notice about the file ***file2-phy***?
7) Create a symbolic link to the file ***file2*** in the directory ***Courses*** under the name ***file2-sym***.
8) View all the information about the files ***file2-phy*** and ***file2-sym***. What differences do you notice?
9) Try changing the access permissions to the file ***file2-sym***. What do you find?
10) Change the access permission for the directory ***Dir1*** so that you no longer have access to it. Try displaying the contents of ***file2-phy*** and ***file2-sym***. What do you notice?
11) Change the access rights of the directory ***Dir1*** again to gain access to it. Move the file ***file2*** into the directory ***Dir1***. Try displaying the contents of the files ***file2-phy*** and ***file2-sym***. What do you notice?
12) Move the file ***file2*** back into the directory ***Dir3***. Try again to display the contents of the files ***file2-phy*** and ***file2-sym***. Remove the file ***file2***. What do you notice?

Exercise n° 2: Creating links on directories

1) Re-create the file ***file2*** in the directory ***Dir3*** with the same contents as in exercise 1?
2) Now take a closer look at the directory ***Dir2***. How many links are there in this directory?
3) In the directory ***Dir2***, create a sub-directory called ***Dir21***. How many links are there in this directory? Explain.
4) Create a physical link called ***Dir3-phy*** from the directory ***Dir3*** to the directory ***Courses***. What do you notice?

5) Create a symbolic link called ***Dir3-sym*** from the directory ***Dir3*** to the directory ***Courses***.

6) Create a symbolic link called ***Dir3-sym-sym*** from the directory ***Dir3-sym*** into the directory ***Dir2***.

7) Move to the directory ***Dir3*** from the directory "***Courses***", then move up to the parent directory using the command ***cd..***.

8) What happens if you use the command ***ls-Ral*** on ***Dir3-sym***? and on ***Dir3-sym-sym***?

Exercise n° 3: **i-nodes (index nodes)**

1) Display the i-node number of the file ***file1***. Copy the file ***file1*** into the directory ***Dir21***. What is its i-node number?

2) Change the name of this file to ***file2*** (***file1*** placed in ***Dir21)***. Does the i-node number change?

3) In the directory ***Dir3***, create a file named ***file3***. In the directory ***Courses***, create a symbolic link and a physical link to this new file, naming them ***file3-sym*** and ***file3-phy*** respectively.

4) Display the i-node number of the file ***file3-sym*** by listing the entire directory ***Courses*** and then listing just the file ***file3-sym***. What do you notice?

5) Compare the i-node numbers between the file ***file3*** in the directory ***Dir3***, ***file3-sym*** and ***file3-phy***. What do you notice?

6) Now look at the i-node number of the root and your home directory.