

Université de Cergy-Pontoise

RAPPORT

pour le projet Systèmes d'Exploitation
Licence d'Informatique troisième année

sur le sujet

Système de gestion de fichiers simple

rédigé par

GOUGEROT Elisa & PAUQUET Jean-Philippe & FERNANDES Raphaël



Mars 2019

Table des matières

1	Introduction	2
1.1	L'objectif du projet	2
1.2	Les outils	2
1.3	Livraisons attendues	2
2	Choix de structures et algorithmes	3
2.1	Gestion de l'espace mémoire	3
2.1.1	Structure Inode	4
2.1.2	Structure Tête de fichier	5
2.1.3	Structure Liste d'adresses	5
2.1.4	Structure Liste de blocs mémoire	6
2.1.5	Les constantes	6
2.2	Opérations de base	7
2.2.1	Création d'un fichier/répertoire	7
2.2.2	Copie d'un fichier/répertoire	8
2.2.3	Renommer d'un fichier/répertoire	9
3	Software installation and user's guide	10

Table des figures

1	Architecture de gestion de l'espace mémoire	3
2	Structure Inode	4
3	Structure Tête de fichier	5
4	Structure Liste d'adresses	5
5	Structure Liste de bloc mémoire	6
6	Création d'un répertoire	7
7	Copie d'un dossier	8
8	Renommage d'un fichier	9

1 Introduction

Notre équipe Le groupe est composé de Fernandes Raphaël, Gougerot Elisa et Pauquet Jean-Philippe.

1.1 L'objectif du projet

Le projet consiste à réaliser un petit gestionnaire de fichiers. Notre partition permet la réalisation de commandes de bases depuis la console ainsi qu'une gestion de l'espace mémoire.

1.2 Les outils

- Atom pour la rédaction des fichiers
- GitHub pour l'hébergement et le partage des fichiers
- Overleaf pour la rédaction du rapport
- Doxygen pour générer une documentation HTML

1.3 Livraisons attendues

Le dossier du projet au format ".zip" contenant :

- Les codes sources des programmes
- Un makefile pour automatiser la production de l'exécutable
- Le rapport de projet (PDF) :
 - Guide d'installation et d'utilisation du logiciel
 - Document décrivant les choix de structures de données et d'algorithmes

2 Choix de structures et algorithmes

Dans cette section, nous présentons la réalisation de notre projet. Comment nous avons g  rer l'espace m  moire de notre syst  me mais   galement quelles op  rations de bases nous avons rendu r  alisable.

2.1 Gestion de l'espace m  moire

Nous avons fait le choix de fixer la taille de la m  moire    1Mo afin de pouvoir y faire des essais probants. La taille de chaque bloc m  moire est de 1ko. Lors de l'  criture d'un fichier de 7ko par exemple, nous utiliserons donc 7 blocs. Voici un sch  ma de notre architecture dont nous allons par la suite d  tailler tous les   l  ments :

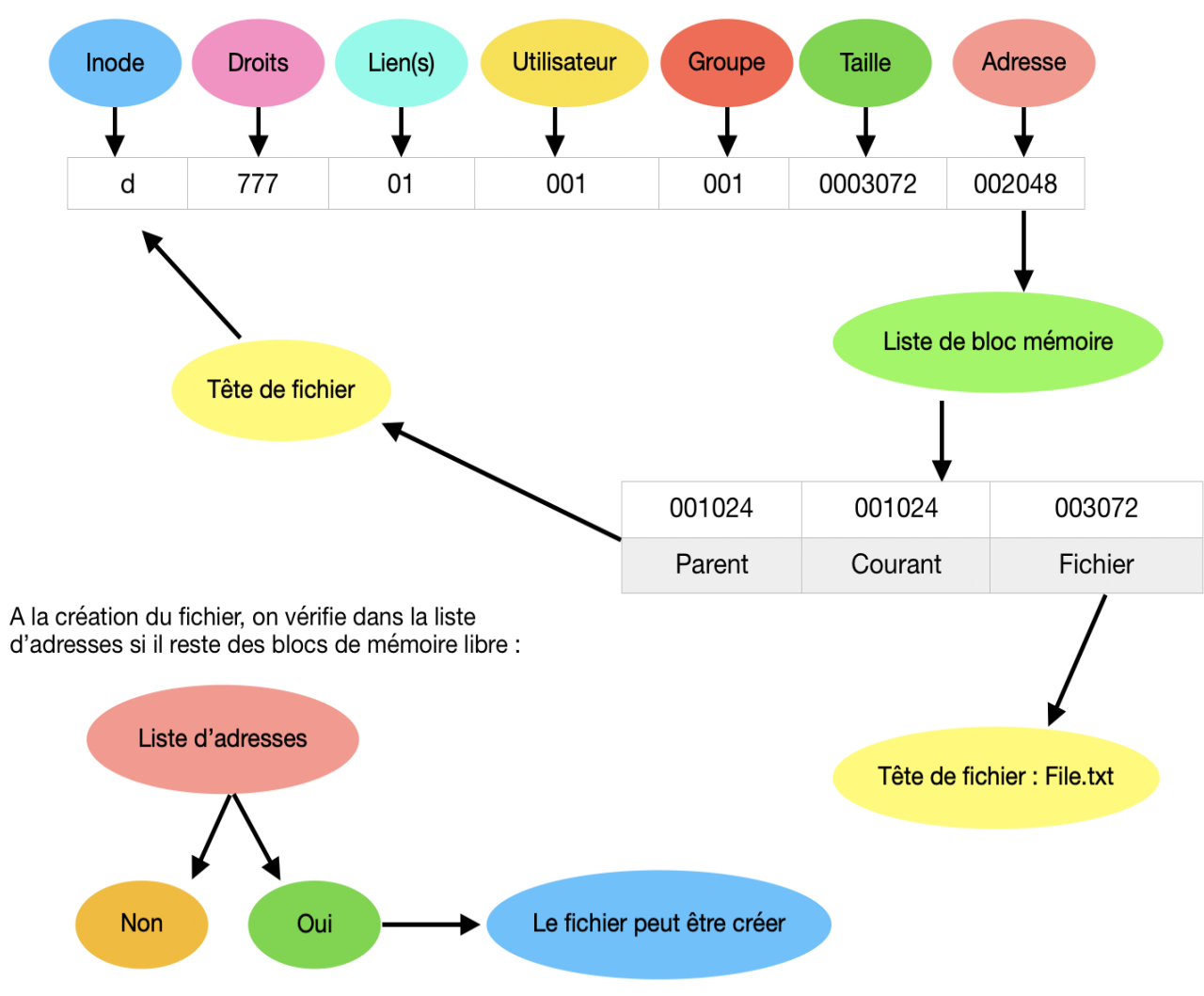


FIGURE 1 – Architecture de gestion de l'espace m  moire

Nous utilisons 4 types de structure diff  rentes, l'inode, la t  te de fichier, la liste d'adresses et la liste de blocs m  moire.

2.1.1 Structure Inode

Notre première structure est l'inode. Il correspond à un numéro unique qui est attribué à chaque fichier et qui est stocké dans la liste de bloc mémoire. L'inode est donc un pointeur vers une case de la mémoire.

Voici le schéma de notre structure Inode :

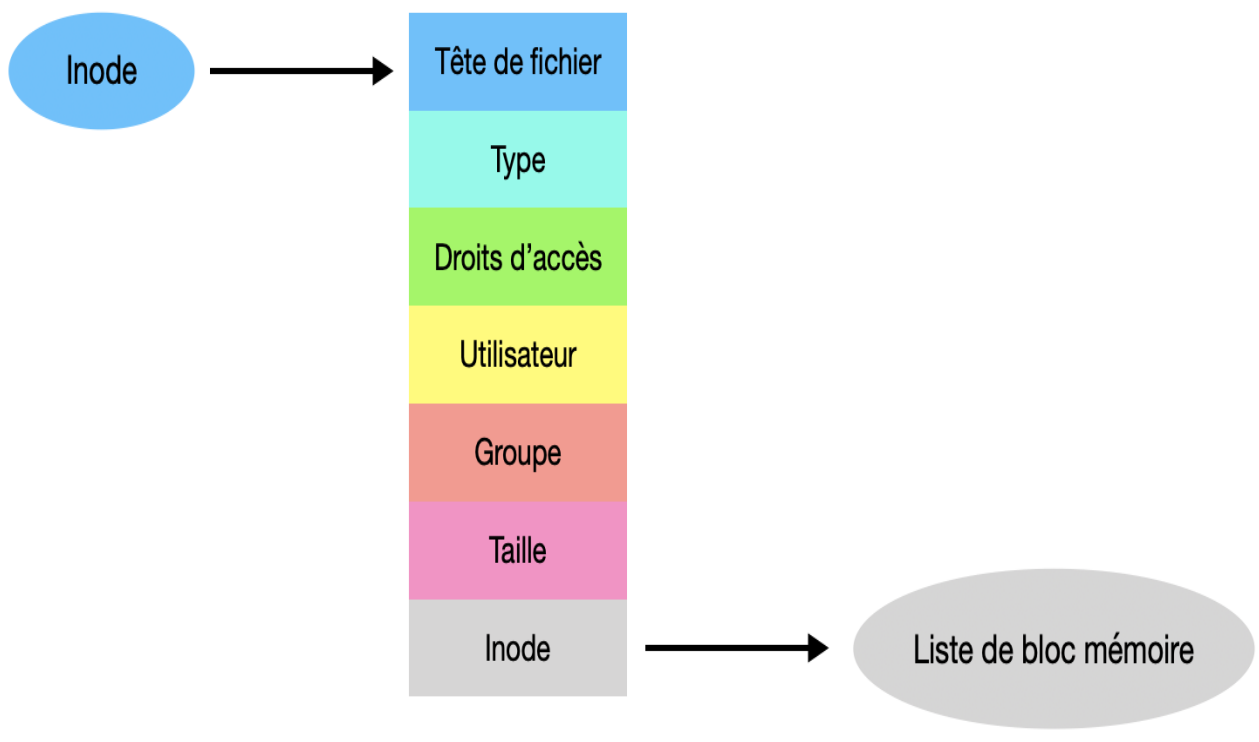


FIGURE 2 – Structure Inode

2.1.2 Structure Tête de fichier

Notre seconde structure est la tête de fichier. Elle permet de stocker les informations d'un fichier. Ainsi chaque fichier contient directement toutes les informations qui lui sont propre notamment :

- Le type de fichier (fichier ou repertoire)
- Les droits d'accès pour un utilisateur ou un groupe
- Le nombre de lien
- La taille du fichier
- L'adresse du fichier
- Le nom du fichier

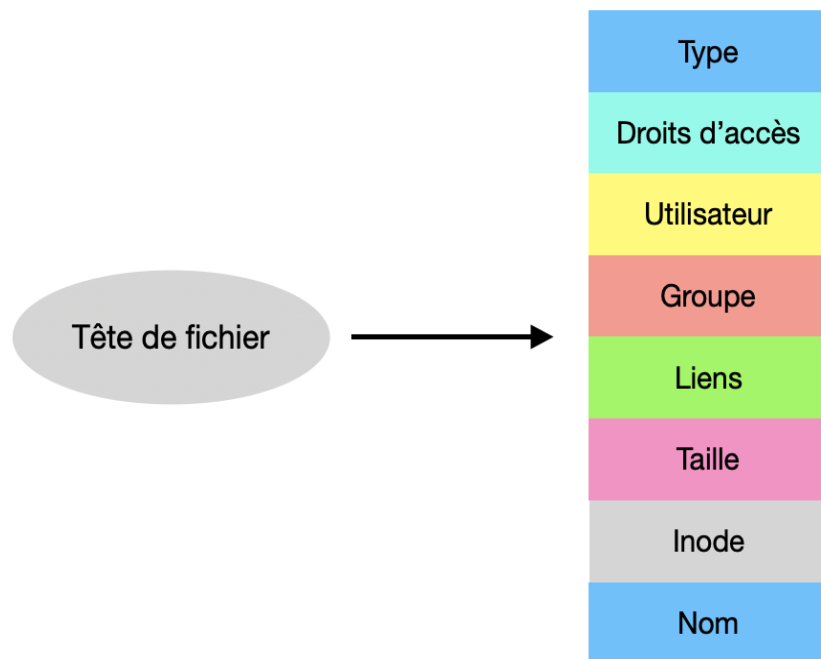


FIGURE 3 – Structure Tête de fichier

2.1.3 Structure Liste d'adresses

Notre troisième structure est la liste d'adresses. Elle correspond à un tableau d'entier (0 ou 1) et elle permet de savoir si le bloc mémoire concerné est libre ou occupé. Elle détermine donc la création d'un fichier. Il y a autant d'adreeses dans la liste d'adresses que de blocs dans la mémoire.

Voici le schéma de notre structure liste d'adresses :

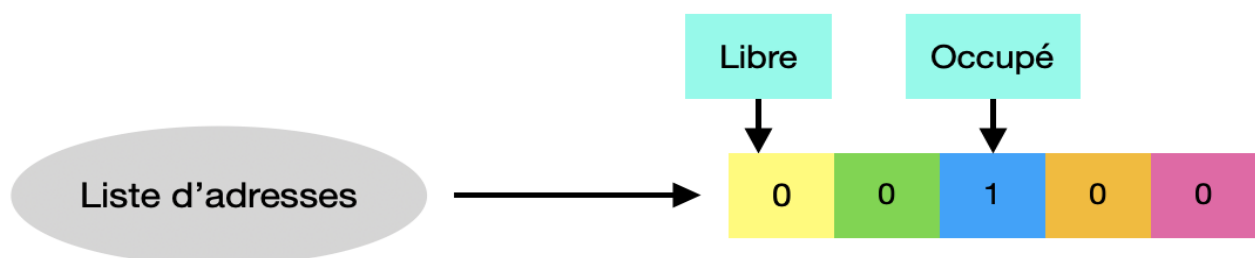


FIGURE 4 – Structure Liste d'adresses

2.1.4 Structure Liste de blocs mémoire

Notre dernière structure est la liste de blocs mémoire. Elle permet de stocker en mémoire, sous la forme de bloc, une liste d'inode. Chaque inode pointe vers un bloc mémoire.

Voici le schéma de notre structure liste de blocs mémoire :

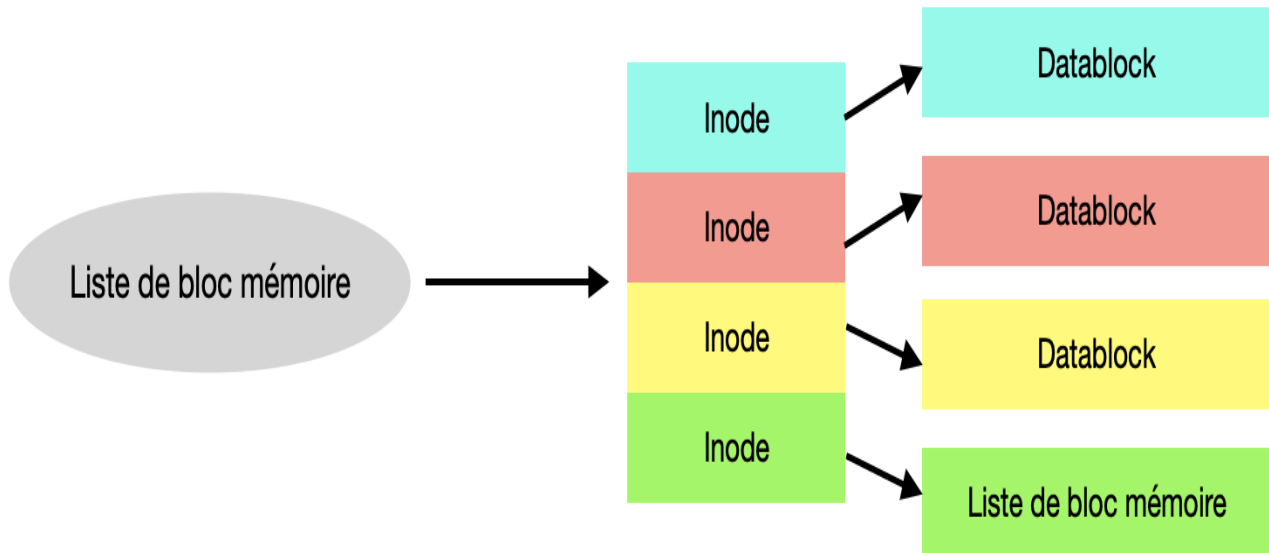


FIGURE 5 – Structure Liste de bloc mémoire

2.1.5 Les constantes

Nous avons choisi arbitrairement la taille totale de la mémoire et la taille d'un bloc. Nous traitons 128 blocs (ce qui correspond à la taille de la liste d'adresses) mémoire de 1024 octet.

Les constantes utilisées pour la mémoire :

- La taille total de la mémoire : 1048576
- La taille d'un bloc : 1024
- Le nombre de caractère pour écrire un inode : 6
- Le nombre d'inode dans la liste de bloc mémoire : 128

Les constantes utilisées pour la tête de fichier :

- Le nombre de type de fichier possible : 1
- Le nombre de caractères pour les droits d'accès : 3
- Le nombre de lien : 2
- Le nombre de caractères pour les droits d'accès pour un utilisateur : 3
- Le nombre de caractères pour les droits d'accès pour un groupe : 3
- La taille d'un fichier : 7
- Le nombre de caractère pour écrire le nom du fichier : 30

2.2 Opérations de base

2.2.1 Création d'un fichier/répertoire

Pour la création d'un fichier ou d'un répertoire nous avons fait le choix d'utiliser l'algorithme suivant :

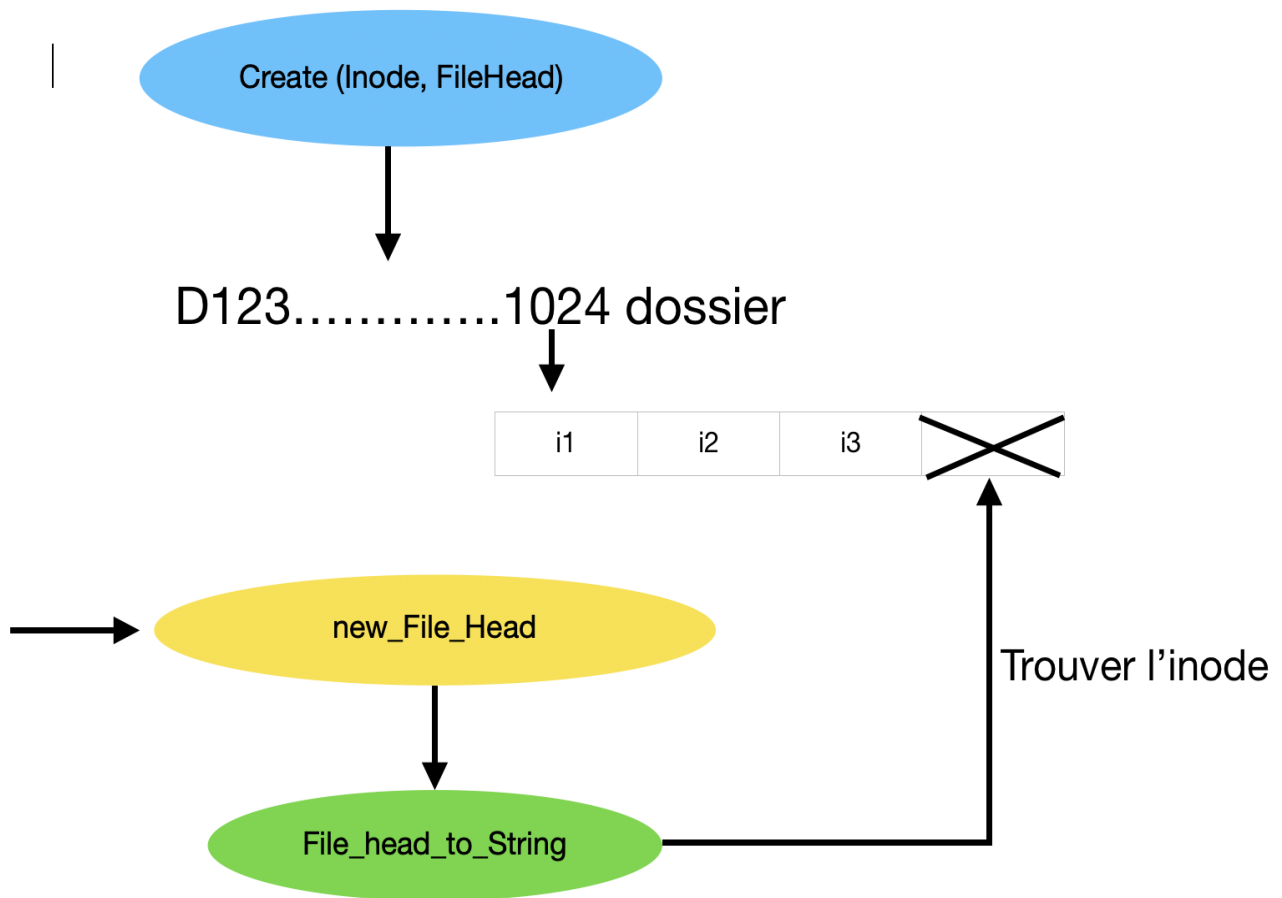


FIGURE 6 – Création d'un répertoire

Pour la création d'un fichier/répertoire, on crée une tête de fichier avec le nom que l'utilisateur a choisi et on va interroger la liste d'adresses pour trouver l'adresse d'un bloc libre. On ajoute à la liste de blocs du dossier courant l'adresse libre. A cette adresse libre, on écrit dans la mémoire la tête de fichier.

2.2.2 Copie d'un fichier/répertoire

Pour la copie d'un fichier ou d'un répertoire nous avons fait le choix d'utiliser l'algorithme suivant :

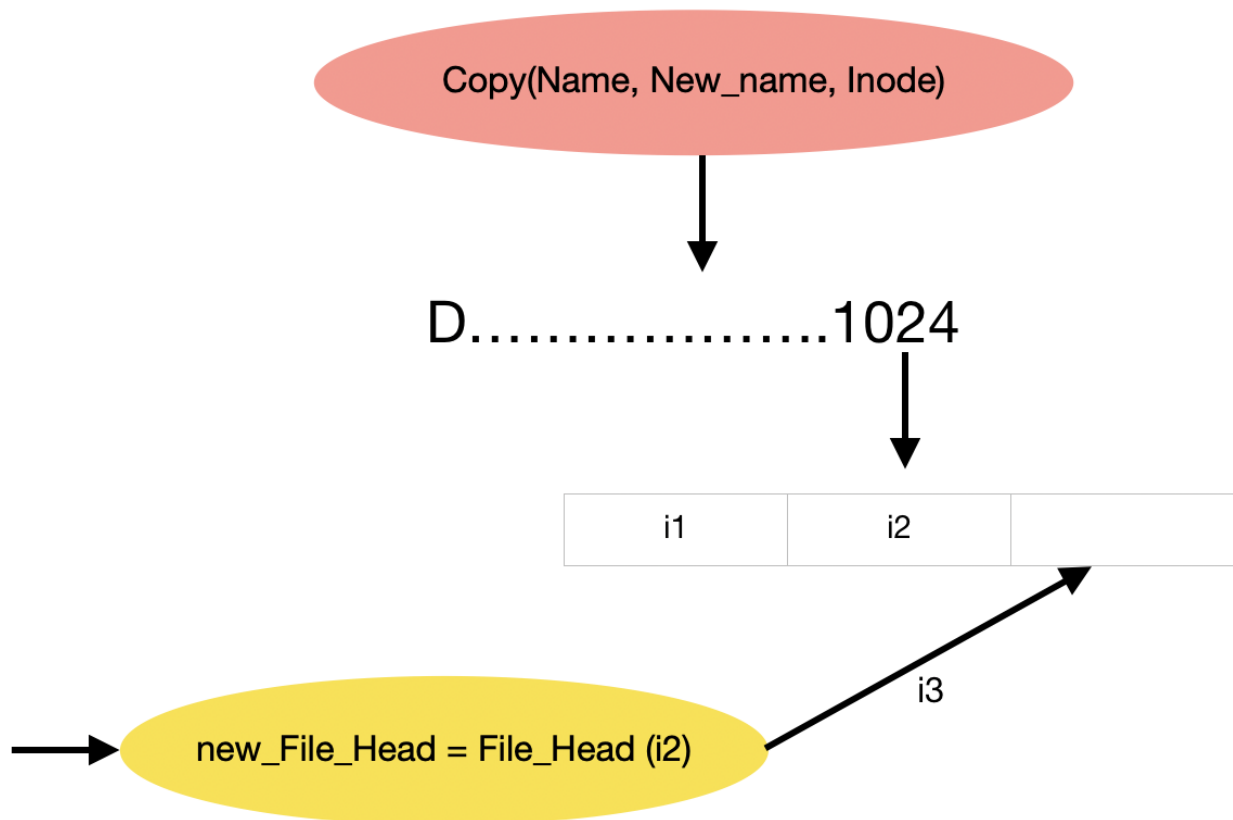


FIGURE 7 – Copie d'un dossier

Pour copier un fichier/répertoire, on va chercher grâce à son nom, son inode dans le répertoire courant. On crée une tête de fichier équivalente au fichier/répertoire qu'on veut copier. On renomme cette tête de fichier avec le nouveau nom de fichier/répertoire. On va chercher dans la liste d'adresses une adresse de bloc disponible. Ensuite, on ajoute à la liste de blocs du dossier courant l'adresse disponible. Enfin, on écrit à cette adresse, la tête de fichier que l'on a créé.

2.2.3 Renommer d'un fichier/répertoire

Pour le renommage d'un fichier ou d'un répertoire nous avons fait le choix d'utiliser l'algorithme suivant :

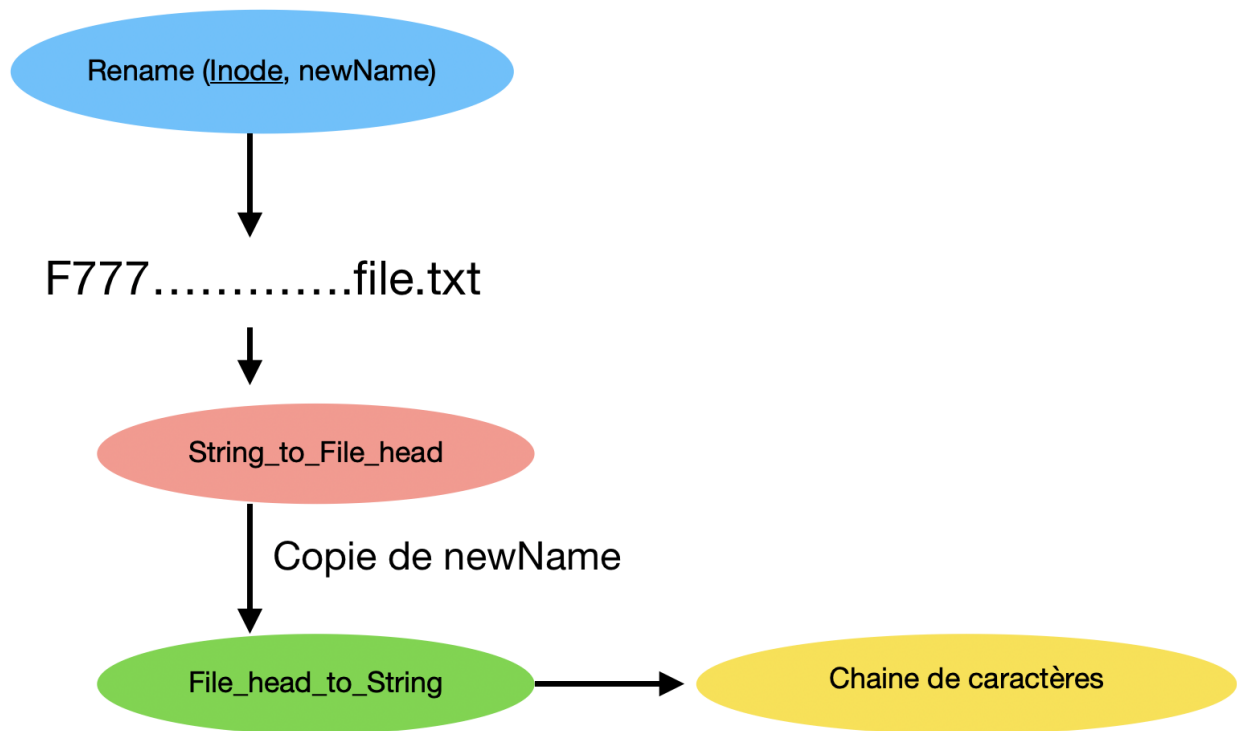


FIGURE 8 – Renommage d'un fichier

Pour le renommage d'un fichier/répertoire, on va chercher la tête de fichier grâce au nom du fichier que l'on veut renommer. On remplace le nom du fichier dans cette tête de fichier par le nouveau nom.

3 Software installation and user's guide

For the software installation, in the terminal, we need to go in the folder that has the main, and then we need to type "make". It will compile the project and run it.

To use the software, several possibilities are available. You can type on the console :

- help : to get access to the user's manual
- q : to leave the program
- create : to create a file, we have to type create and then type the file's name and the file's type (f or d).
- file's copy : to copy a file, we have to type "cp", and then the file's name and the new file's name that we want.
- file's rename : to rename a file we have to type "rename", the former name and then the new name that we want.