



R 2.04

2023 - 2024

Communication et fonctionnement bas niveau

TD N°4

« Programmation de commandes de gestion du Système de Fichiers »



ANNE Jean-François
D'après le TD de F. BOURDON

Le but de ce TD est de se familiariser avec l'architecture bas niveau système et réseau.

« Programmation de commandes de gestion du Système de Fichiers »

Notions vues dans ce TD :

Manipulation des structures internes du SGF (stat, direct, DIR), récursivité.

Nombre de séance de **2h00** prévu pour faire ce TD : **1**.


Prochain TD : Allocation dynamique de mémoire (malloc...).

PS : Les parties correspondant à du travail à faire sont toutes en italiques ; le restant étant du complément au cours.

1°). Écriture d'un programme d'affichage pour lister le contenu d'un répertoire de manière récursive.

Le but de cet exercice est de manipuler en langage « C » les structures internes du système de fichiers d'UNIX. Pour cela nous allons écrire un programme « Rdir.c » qui effectuera un « ls » récursif en affichant pour chaque élément :

N° d'i-noeud, nom, taille, uid et gid des propriétaires, type (répertoire ou fichier).

 *Écrire et lancer ce programme « Rdir.c » en vous inspirant des fichiers « dir.c » et « status.c ».*

« dir.c »

```
#include <stdio.h>
#include <sys/dir.h>

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

    /* la structure DIR permet de récupérer des informations sur un répertoire
    */
    DIR* dir;

    /* la structure «direct» (définie dans "/usr/include/sys/dir.h") synonyme
    de la structure «dirent», */
    /* contient des informations sur un élément du répertoire */
    /* (le détail d'une entrée de répertoire, c'est-à-dire le "i-noeud" du
    fichier et son nom) */
    /* la structure "dirent" se trouve dans le fichier
    "/usr/include/bits/dirent.h".
    struct dirent
```

```

{
    #ifndef __USE_FILE_OFFSET64
    __ino_t d_ino;
    __off_t d_off;
    #else
    __ino64_t d_ino;
    __off64_t d_off;
    #endif
    unsigned short int d_reclen;
    unsigned char d_type;
    char d_name[256];
};
*/
struct direct* dirdet;
char* chem;

/* controle des parametres d'appel */
if(argc != 2) {
printf("erreur de syntaxe d'appel !!!\n");
exit(0);
}
/* recuperation du nom du repertoire */
chem = (char*)malloc(strlen(argv[1]) +1);
strcpy(chem,argv[1]);
/* ouverture du repertoire et controle d'existence */
dir = opendir(chem);
if(dir == NULL) {
printf("erreur opendir de %s !!!\n",chem);
exit(0);
}
/* lecture des donnees du repertoire */
while((dirdet = readdir(dir)) != NULL) {
printf("%s : N ino = %d\tNom = %s\n",chem,dirdet->d_ino,dirdet->d_name);
}
/* liberation memoire */
free(chem);
/* fermeture du repertoire ouvert */
closedir(dir);
}

```

« status.c »

#include <stdio.h>

```
#include <sys/stat.h>

main(int argc, char *argv[]) {
    /* la structure «stat» permet de récupérer les informations contenues dans
    les entêtes (i-noeud) de fichier */
    /* la structure stat se trouve dans le fichier "/usr/i386-glibc20-
    linux/include/asm/stat.h"

    struct stat {
        unsigned short st_dev; // identificateur du périphérique hébergeant le
        fichier.
        unsigned short __pad1;
        unsigned long st_ino; // i-noeud associé au fichier.
        unsigned short st_mode; // mode du fichier (type et droit d'accès).
        unsigned short st_nlink; // nombre de liens absolus vers le fichier.
        unsigned short st_uid; // identificateur de l'utilisateur propriétaire du
        fichier.
        unsigned short st_gid;
        unsigned short st_rdev;
        unsigned short __pad2;
        unsigned long st_size;
        unsigned long st_blksize;
        unsigned long st_blocks;
        unsigned long st_atime; // date du dernier accès.
        unsigned long __unused1;
        unsigned long st_mtime;
        unsigned long __unused2;
        unsigned long st_ctime;
        unsigned long __unused3;
        unsigned long __unused4;
        unsigned long __unused5;
    };
    */

    struct stat st;
    char* chem;
    int rep;

    /* controle des parametres d'appel */
    if(argc != 2) {
        printf("erreur de syntaxe d'appel !!!\n");
        exit(0);
    }

    /* recuperation du nom du repertoire */
    chem = (char*)malloc(strlen(argv[1]) +1);
    strcpy(chem,argv[1]);
```

```
/* recuperation des infos du i_noeud par stat */
rep = stat(chem,&st);
if(rep == 0) {
printf("%s : ino = %d\ttaille = %d\t",chem,st.st_ino,st.st_size);
if(S_ISDIR(st.st_mode)) printf(" DIRECTORY ");
if(S_ISREG(st.st_mode)) printf(" FICHER ");
if(S_ISLNK(st.st_mode)) printf(" LIEN ");
printf("owner = %d\tgrpe = %d\n",st.st_uid,st.st_gid);
}
else {
printf("erreur sur stat de %s !!!\n");
exit(0);
}
/* liberation memoire */
free(chem);
}
```

I. Webographie :

- <https://bourdon.users.info.unicaen.fr/cours/IUT-1A/index.html>
-