

# Rappel sur la manipulation des fichiers

## 1 Primitives

### 1.1 Les opérations de base

```
int open(const char *path, int oflag, /* mode_t mode */...);
int close(int fildes);
ssize_t write(int fildes, const void *buf, size_t nbyte);
ssize_t read(int fildes, void *buf, size_t nbyte);
```

### 1.2 Manipulation de la table des fichiers

```
int fcntl(int fildes, int cmd, /* arg */ ...);
```

Cette fonction réalise des opérations de contrôle sur des descripteurs de fichiers. C'est une fonction générale appliquant une commande à un fichier avec, éventuellement, un argument. On s'en sert pour consulter ou modifier les caractéristiques des fichiers (mode d'ouverture, entree dans la table des descripteurs, verrous...)

### 1.3 Manipulation des descripteurs

```
int dup(int fildes); <=> fcntl(fildes, F_DUPFD, 0);
int pipe(int fildes[2]);
```

### 1.4 Modification de la position courante

```
off_t lseek(int fildes, off_t offset, int whence);
```

### 1.5 Modification des i-noeuds et des catalogues

```
/* Creation d'un i-noeud (limitee) */
int mknod(const char *path, mode_t mode, dev_t droits);
/* Creation d'un catalogue */
int mkdir(const char *path, mode_t mode);
/* Suppression d'un catalogue */
int rmdir(const char *path);
/* Changement de catalogue courant */
int chdir(const char *path);
```

## 1.6 Consultation d'un i-noeud

```
int stat(const char *path, struct stat *buf);
int lstat(const char *path, struct stat *buf);

avec
struct stat
{
    mode_t    st_mode;    /* Mode du noeud */
                        /* bits 15 a 12 : type */
                        /* bits 11 a 0 : mode d'accès */
    ino_t     st_ino;    /* Numero du noeud */
    dev_t     st_dev;    /* numero ressource contenant un catalogue */
                        /* ayant une reference du noeud */
    dev_t     st_rdev;    /* numero ressource pour un fichier special */
    nlink_t   st_nlink;  /* Nombre de liens */
    uid_t     st_uid;    /* numero du proprietaire */
    gid_t     st_gid;    /* Numero du groupe */
    off_t     st_size;   /* Taille du fichier en octets */
    time_t    st_atime;  /* Date du dernier accès au fichier */
    time_t    st_mtime;  /* Date de derniere modif */
    time_t    st_ctime;  /* Date de derniere modif de caracteristique */
    long      st_blksize; /* Taille de bloc d'E/S voulue */
    blkcnt_t  st_blocks; /* Nombre de blocs de 512 octets alloués */
}
```

## 1.7 Modification des caractéristiques d'un i-noeud

```
/* Changement mode d'accès */
int chmod(const char *path, mode_t mode);
/* Changement proprietaire */
int chown(const char *path, uid_t owner, gid_t group);
```

## 1.8 Manipulation des liens

```
/* Lien physique */
int link(const char *existing, const char *new);
/* Lien symbolique */
int symlink(const char *name1, const char *name2);
/* Supprime une entree dans le catalogue */
int unlink(const char *path);
```

## 2 Fonctions E/S avec tampon

Ce sont les fonctions d'entrée/sortie standard de la bibliothèque `stdio.h`.

### 2.1 Ouverture/Fermeture

```
/* Ouverture */
FILE *fopen(const char *filename, const char *mode);
/* Fermeture */
int fclose(FILE *stream);
/* Passerelle descripteur int --> descripteur FILE */
FILE *fdopen(int fildes, const char *mode);
/* Passerelle descripteur FILE --> descripteur int */
FILE * fd ;
fd->_file ...
```

### 2.2 Lecture/Ecriture d'un caractère

```
/* Lecture */
int fgetc(FILE *stream); /* Fonction */
int getc(FILE *stream); /* Macro */
/* Ecriture */
int fputc(int c, FILE *stream); /* Fonction */
int putc(int c, FILE *stream); /* Macro */
```

### 2.3 Lecture/Ecriture d'une chaîne de caractères

```
char *fgets(char *s, int n, FILE *stream);
int fputs(const char *s, FILE *stream);
```

### 2.4 Déplacement du pointeur dans le tampon

```
int fseek(FILE *stream, long offset, int whence);
long ftell(FILE *stream);
```

### 2.5 Lecture/Ecriture formatée

```
int fprintf(FILE *stream, const char *format, /* args*/ ...);
int fscanf(FILE*stream, const char *format, ...);
```

### 2.6 Entrée/Sortie en mémoire

```
int sprintf(char *s, const char *format, /* args*/ ...);
int sscanf(const char *s, const char *format, ...);
```

### 2.7 Lecture/Ecriture en binaire

```
size_t fread(void *ptr, size_t size, size_t nitems, FILE *fd);
```

```
size_t fwrite(const void *ptr, size_t size, size_t nitems, FILE *fd );
```

### 3 Fonctions de consultation des répertoires

Elles font partie de l'interface POSIX. Ce sont les fonctions de la bibliothèque `dirent.h`. Elles permettent de lire des répertoires, distants ou non, indépendamment de leur support physique. Elles s'appuient sur la "structure générique" :

```
typedef struct dirent {
    ino_t      d_ino;    /* numero d'index de fichier */
    off_t      d_off;    /* longueur d'enregistrement */
    unsigned short d_reclen; /* longueur du nom dans d_name */
    char       d_name[1]; /* nom du fichier */
} dirent_t;
```

Les fonctions sont :

```
/* Ouverture d'un repertoire */
DIR *opendir(const char *dirname);
/* Lecture d'une entree */
struct dirent *readdir(DIR *dirp);
/* "Rembobinage" */
void rewinddir(DIR *dirp);
/* Fermeture d'un repertoire */
int closedir(DIR *dirp);
```