LINUX – LE NOYAU
ACCÈS AUX FICHIERS

MINISTÈRE
DE L'ÉDUCATION NATIONALE,
DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE

www.yncrea.fr

Descripteur de fichiers

- Les entrées/sorties sont identifiés par un descripteur de fichier (nombre entier)

- Un programme hérite de trois fichiers ouverts dont les descripteurs sont :
  - **0** -> pour l'entrée standard (clavier) ou stdin
  - **1** -> pour la sortie standard (écran, terminal) ou stdout
  - **2** -> pour la sortie d'erreur standard (écran, terminal) ou stderr

- Quand le système ouvrira d'autres fichiers :
  - les descripteurs prendront pour valeur 3, 4…

Création d'un fichier

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <limits.h>

main()
{
        int fd;
        fd = creat( "DATA", 0777);
        printf("File descriptor: %d\n", fd);
        …
}
```

Nom du fichier

Mode

Les modes de création de fichiers

User	Group	Others
r w x	r w x	r w x
1 1 1	1 1 1	1 1 1
7	7	7

S_IRWXU|S_IRWXG|S_IRWXO

1 0 0	0 0 0	0 0 0
4	0	0

S_IRUSR

S_IRWXU
00700 user (file owner) has read, write and execute permission
S_IRUSR (S_IREAD)
00400 user has read permission
S_IWUSR (S_IWRITE)
00200 user has write permission
S_IXUSR (S_IEXEC)
00100 user has execute permission
S_IRWXG
00070 group has read, write and execute permission
S_IRGRP
00040 group has read permission
S_IWGRP
00020 group has write permission
S_IXGRP
00010 group has execute permission
S_IRWXO
00007 others have read, write and execute permission
S_IROTH
00004 others have read permission
S_IWOTH
00002 others have write permission
S_IXOTH
00001 others have execute permission

Ouverture et fermeture d'un fichier

```
int open(path, flags) char *path;
        int flags;

int open(path, flags, mode)  char *path;
        int flags;
        int mode;

close(fd)
        int fd;
```

Retourne un descripteur de fichier (-1 si erreur)

Ouvre un fichier selon le mode spécifié par **flags** :

O_RDONLY Open for reading only
O_WRONLY Open for writing only
O_RDWR Open for reading and writing
O_APPEND Append on each write
O_CREAT Create file if it does not exist
O_TRUNC Truncate size to 0
O_EXCL Error if create and file exists
O_BLKINUSE Block if file is in use
O_BLKANDSET Block if file is in use; then set in use
O_FSYNC Do file writes synchronously
…

Lecture et écriture de fichier

```
int  read(fd, buf, nbytes)
        int fd;
        char *buf;
        int nbytes;
```

Retourne le nombre d'octet(s) lu(s)
(-1 si échec !)

- Descripteur de fichier
- Adresse mémoire
- Nombre d'octet(s) à lire

```
int  write(fd, buf, nbytes)
        int fd;
        char *buf;
        int nbytes;
```

Retourne le nombre d'octet(s) écrit(s)
(-1 si échec !)

- Descripteur de fichier
- Adresse mémoire
- Nombre d'octet(s) à écrire

Exemple

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/uio.h>
#include <string.h>
#include <limits.h>

main()
{
      int fd;
      int cc;
      char *buf = "#file: DATA - date: Tue Sep 22 16:03:05 WET 1992 author:root\n";

      if ( ( fd = open( "DATA", O_CREAT|O_WRONLY, S_IRWXU )  ) !=  -1) {
            if ( ( cc = write( fd, buf, strlen(buf) ) ) != strlen(buf) ) {
                  printf("Erreur d'écriture dans le fichier\n");
            }
            printf("Nombre de caractères écrits : %d\n", cc);
            close(fd);
      }
}
```

Exemple

Positionnement dans un fichier

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

Positionner la « tête » de lecture/écriture dans un fichier

(retourne le nouvel emplacement, mesuré en octets depuis le début du fichier si succès, -1 si échec !)

Place la tête de lecture/écriture à la position **offset** dans le fichier associé au descripteur **fd** en suivant la directive **whence**

0 : début du fichier
1 : position courante
2 : fin du fichier

Exemple

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/uio.h>
#include <string.h>
#include <limits.h>
main()
{
        int fd, cc;
        char *buf1  = "oui oui";
        char *buf2 = "stit";
        char *buf3 =  "ou non";
        if ( (fd=open("DATA", O_CREAT|O_WRONLY, 0700) ) != -1) {
                cc = write( fd, buf1, strlen(buf1) );
                printf("Nombre de caractères écrit : %d\n", cc);
                lseek(fd, 2, 0);
                write( fd, buf2, strlen(buf2) );
                lseek(fd, 3, 2);
                write( fd, buf3, strlen(buf3) );
                close(fd);
        }
}
```

o	u	i		o	u	i									
o	u	s	t	i	t	i									
o	u	s	t	i	t	i				o	u		n	o	n

Exemple

Autres fonctions…

chdir (…)
rmdir (…)
chmod (…)
chown (…)
chroot (…)
…