

Programmation système.

Séance à distance du 02/11/2020

Correction cat

```
#include<...>

int main()
{
    int nbOct;
    char buffer[100];

    while( (nbOct = read( 0 , buffer , sizeof(buffer) )) > 0 )
        if( write( 1, buffer , nbOct ) != nbOct) perror(« Erreur write »);

    if(nbOct<0) perror(« Error read »);

    return(0);
}
```

Exercice

- Prompt\$./monCp fichierSrc.txt fichierDst.txt
- Ce que fait le programme
 - Il lit le fichierSrc.txt
 - Tout ce qui est lu est écrit dans fichierDst.txt
- Pour ouvrir un fichier: open(2)
 - Mis en place du contexte permettant au SE de manipuler le fichier (structures)
 - Renvoi un identifiant au processus permettant à celui-ci de lire/écrire sur le fichier

Open(2)

- int open(char* cheminDuFichier, int flags);
- int open(char* cheminDuFichier, int flags, mode_t mode);
- **Le descripteur de fichiers correspondant à l'ouverture est renvoyé par open.**

flags: O_RDONLY, O_WRONLY, O_RDWR, O_TRUNC, O_APPEND, O_CREAT

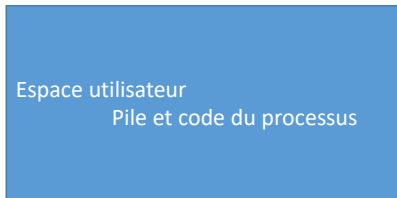
Si en écriture uniquement

O_TRUNC: efface l'ancien contenu.

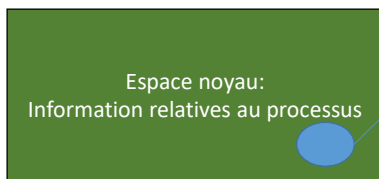
O_APPEND: écrit en mode ajout

O_CREAT: si le fichier n'existe pas on le crée. Dans ce cas, on rajoute le 3^{ème} argument de open (mode) avec les droits du fichier.

Descripteur de fichiers



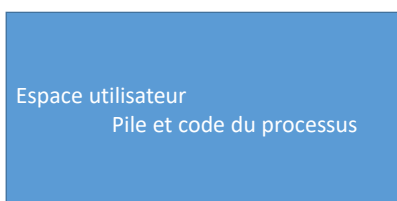
Les descripteurs de fichiers manipulés par le programmeur/processus sont les indices d'un tableau. Il s'agit d'un tableau de pointeurs qui pointent vers les fichiers ouverts.



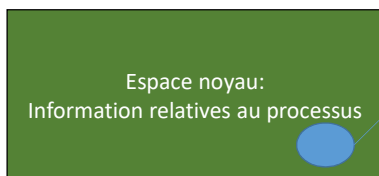
descripteur	
fd[0]	→ Entrée standard
fd[1]	→ Sortie standard
fd[2]	→ Sortie erreur
fd[3]	→ NULL
fd[4]	
fd[5]	

Il y a un contexte par processus: propriétaire, pid, ppid, signaux, tableau des descripteurs de fichiers.

Descripteur de fichiers



Les descripteurs de fichiers manipulés par le programmeur/processus sont les indices d'un tableau. Il s'agit d'un tableau de pointeurs qui pointent vers les fichiers ouverts.



descripteur	
fd[0]	→ Entrée standard
fd[1]	→ Sortie standard
fd[2]	→ Sortie erreur
fd[3]	→ Offset Type d'ouverture (lecture/écriture, etc.)
fd[4]	
fd[5]	

Le fichier ouvert par open

Il y a un contexte par processus: propriétaire, pid, ppid, signaux, tableau des descripteurs de fichiers.

Open: exemples

- `fd=open(« fichier.txt », O_RDONLY);`
 - `read(fd,maChaine,sizeof(maChaine));`
- `fd=open(« /home/toto/Documents/fichier.txt »,O_WRONLY|O_APPEND);`
 - `write(fd,...).`
- `fd=open(« ../fichier.txt »,O_WRONLY|O_CREAT,0621);`

O_RDONLY = 000001 (1 en décimal)
 O_WRONLY = 000010 (2 en décimal)
 O_APPEND = 000100 (4 en décimal)
 O_CREAT = 001000 (8 en décimal)
 O_WRONLY | O_CREAT = 001010
 O_WRONLY | O_CREAT | O_APPEND = 001110

Lecture

Offset=0

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaa
```

La taille de votre tableau fait 2 lignes. Read lit deux lignes et renvoie le nb d'octets lu. Offset=30

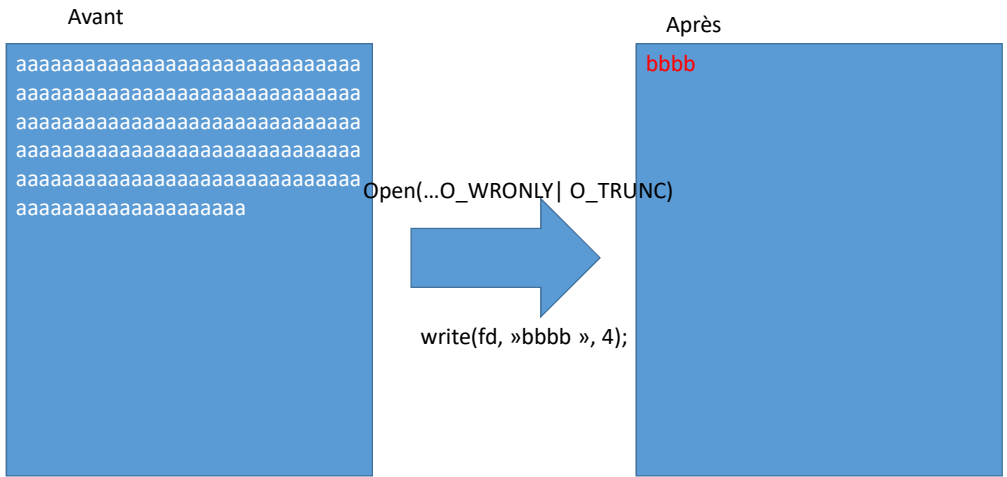
La taille de votre tableau fait 2 lignes. Read lit deux lignes et renvoie le nb d'octets lu. Offset=60

La taille de votre tableau fait 2 lignes. Read lit ce qui reste et renvoie le nb d'octets lu. Offset=77

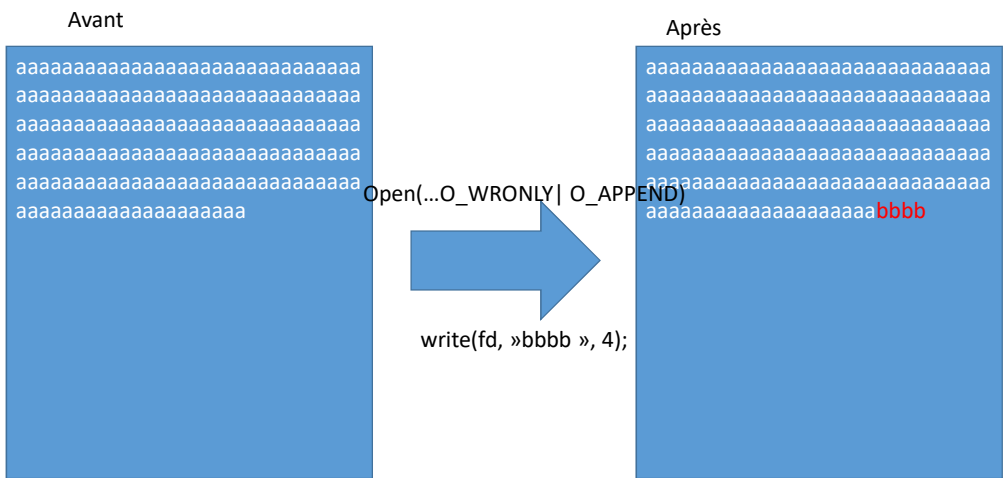
Une fois arrivé en fin de fichier read renvoie 0.

Si un read est fait une nouvelle fois, read renvoie -1 (une erreur).

Ecriture



Ecriture



Ecriture

Avant

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

Open(...O_WRONLY)



write(fd, »bbbb », 4);

Après

```
bbbbaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

Ecrase! Ne décale pas.

Fermeture

- Le fichier est fermé par l'appel à close
 - close(fd)
- Libère les structures mises en place par le système
- Libère le descripteur de fichiers: remet la case du tableau des descripteurs à NULL.