

Module SY5 – Systèmes d'Exploitation

Dominique Poulalhon

`dominique.poulalhon@irif.fr`

Université de Paris (Diderot)

L3 Informatique & DL Bio-Info, Jap-Info, Math-Info

Année universitaire 2021-2022

GESTION DES ENTRÉES/SORTIES

OUVERTURE ET FERMETURE DE FICHIERS

l'accès à un fichier est une action critique \implies appel système

```
int open(const char *pathname, int flags/*, mode_t mode*/);
```

que fait cet appel ?

- il teste si `pathname` est une référence valide et accessible
dans le cas contraire, il renvoie -1 avec positionnement de la variable `errno` selon les valeurs indiquées dans `man 2 open`
- il met en place (en mémoire) les structures nécessaires pour accéder simplement au contenu du fichier
le `descripteur` renvoyé est le point d'accès du processus à ces structures

OUVERTURE ET FERMETURE DE FICHIERS

l'accès à un fichier est une action critique \implies appel système

```
int open(const char *pathname, int flags/*, mode_t mode*/);
```

que fait cet appel ?

- il teste si `pathname` est une référence valide et accessible dans le cas contraire, il renvoie -1 avec positionnement de la variable `errno` selon les valeurs indiquées dans `man 2 open`
 - il met en place (en mémoire) les structures nécessaires pour accéder simplement au contenu du fichier
le `descripteur` renvoyé est le point d'accès du processus à ces structures
-
- `table des descripteurs` *du processus*
 - `table des fichiers ouverts` *du système*
 - `table des i-nœuds` *du système* (en mémoire)

OUVERTURE ET FERMETURE DE FICHIERS

l'accès à un fichier est une action critique \implies appel système

```
int open(const char *pathname, int flags/*, mode_t mode*/);
```

- table des descripteurs *du processus*
- table des fichiers ouverts *du système*
- table des i-nœuds *du système* (en mémoire)

pour libérer les ressources correspondantes :

```
int close(int fd);
```

OUVERTURE ET FERMETURE DE FICHIERS

vous connaissez déjà des descripteurs :

- 0 est le descripteur associé à l'entrée standard ;
- 1 est le descripteur associé à la sortie standard ;
- 2 est le descripteur associé à la sortie erreur standard.

ils sont (en général) *hérités* du processus père et ne nécessitent pas d'ouverture ; nous verrons tout à l'heure comment *changer* les fichiers ouverts associés à ces descripteurs

note : ces trois descripteurs sont aussi définis par des macros dans `unistd.h` : `STDIN_FILENO`, `STDOUT_FILENO` et `STDERR_FILENO`.

LECTURE ET ÉCRITURE DANS DES FICHIERS

```
ssize_t read(int fd, void *buf, size_t count);  
ssize_t write(int fd, const void *buf, size_t count);
```

- `fd` est un descripteur
- `count` est la taille des données à lire ou écrire
- `buf` est l'adresse d'un emplacement mémoire pour stocker les données lues ou lire les données à écrire

la valeur de retour est le nombre d'octets effectivement lus ou écrits – ou -1 en cas d'erreur ; voir `errno` dans ce cas !

. (les types `size_t` et `ssize_t` sont des entiers respectivement non signés et signés pour POSIX.1. Ils servent essentiellement à conserver la compatibilité entre les différentes versions de POSIX)