

Descripteur de fichier et sous-système IPC

SYE

6 - Descripteur Fichier et Tubes

Résumé du document

Definition

Table des matières

1. Descripteur de fichier	2
1.1. Tableau de descripteurs local	2
1.2. Redirection de descripteurs	2
1.3. Strucutre table de descripteurs	2
1.4. Commande dup2	3
2. Sous-système IPC	4
2.1. Propriétés des IPCs	4

1. Descripteur de fichier

Un descripteur de fichier est un identifiant unique vers une ressource de type fichier. Ce descripteur est retourné par l'appel système `open()` et pourra être utilisé avec d'autres appels systèmes effectuant des accès sur le fichier.

1.1. Tableau de descripteurs local

Chaque processus dispose de sa propre table de descripteurs et chaque descripteur pointe vers une entrée dans une table globale appelée **table des fichiers ouverts**. Cette table contient l'ensemble des références vers les ressources en cours d'utilisation (fichiers, tubes, socket réseau, etc...).

Les tables (locales) de descripteurs et la table des fichiers ouverts sont des structures **créées et gérées dans l'espace noyau**.

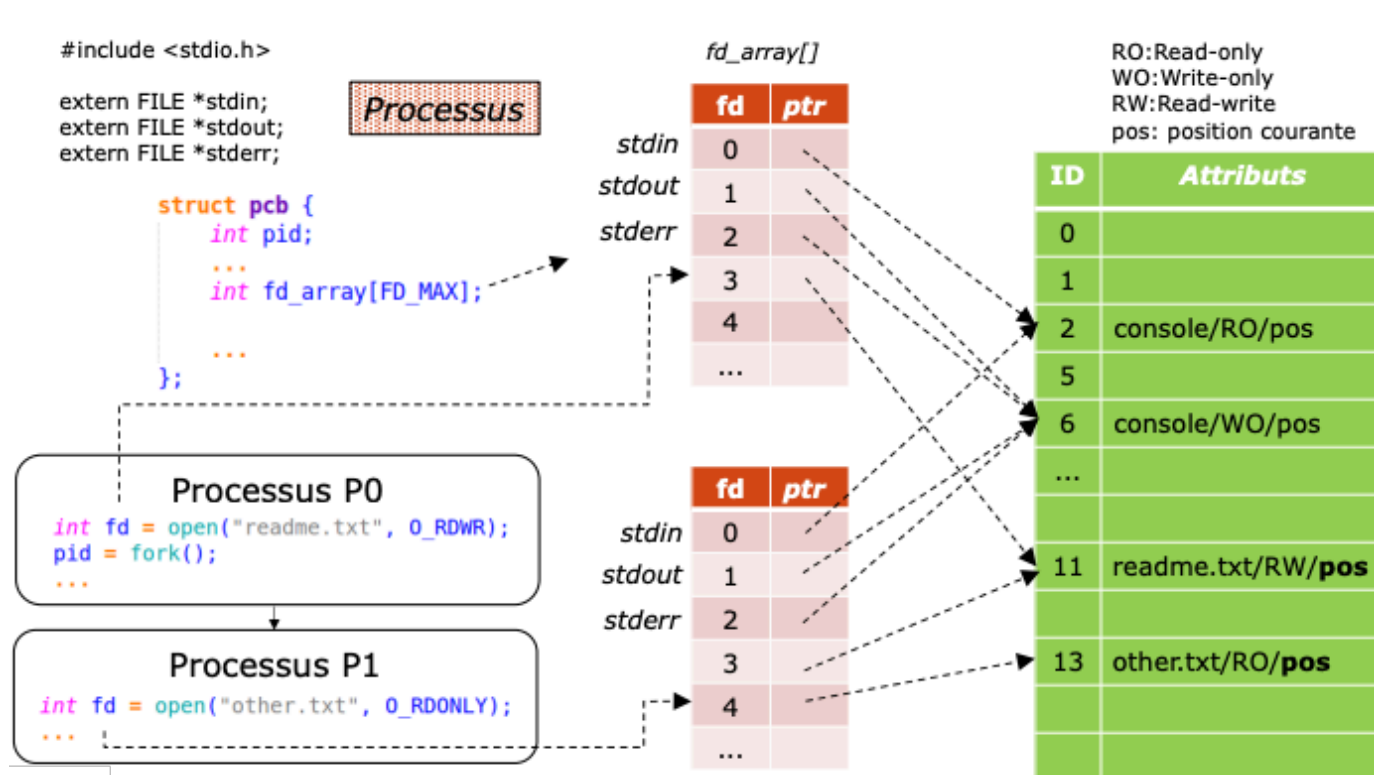
1.2. Redirection de descripteurs

- Un descripteur peut être redirigé vers un autre descripteur.
- La redirection modifie le lien entre un descripteur du tableau local et l'entrée de la table des fichiers ouverts.
- Cette modification permet au descripteur de pointer vers une autre entrée de la table des fichiers ouverts.
- La redirection de descripteur est également appelée "synonyme de descripteur".
- Elle est réalisée via l'appel système `dup2()`.

1.3. Structure table de descripteurs

Dans la table de descripteurs aussi dites `fd_array` les 3 premières lignes sont toujours initialisées aux mêmes valeurs:

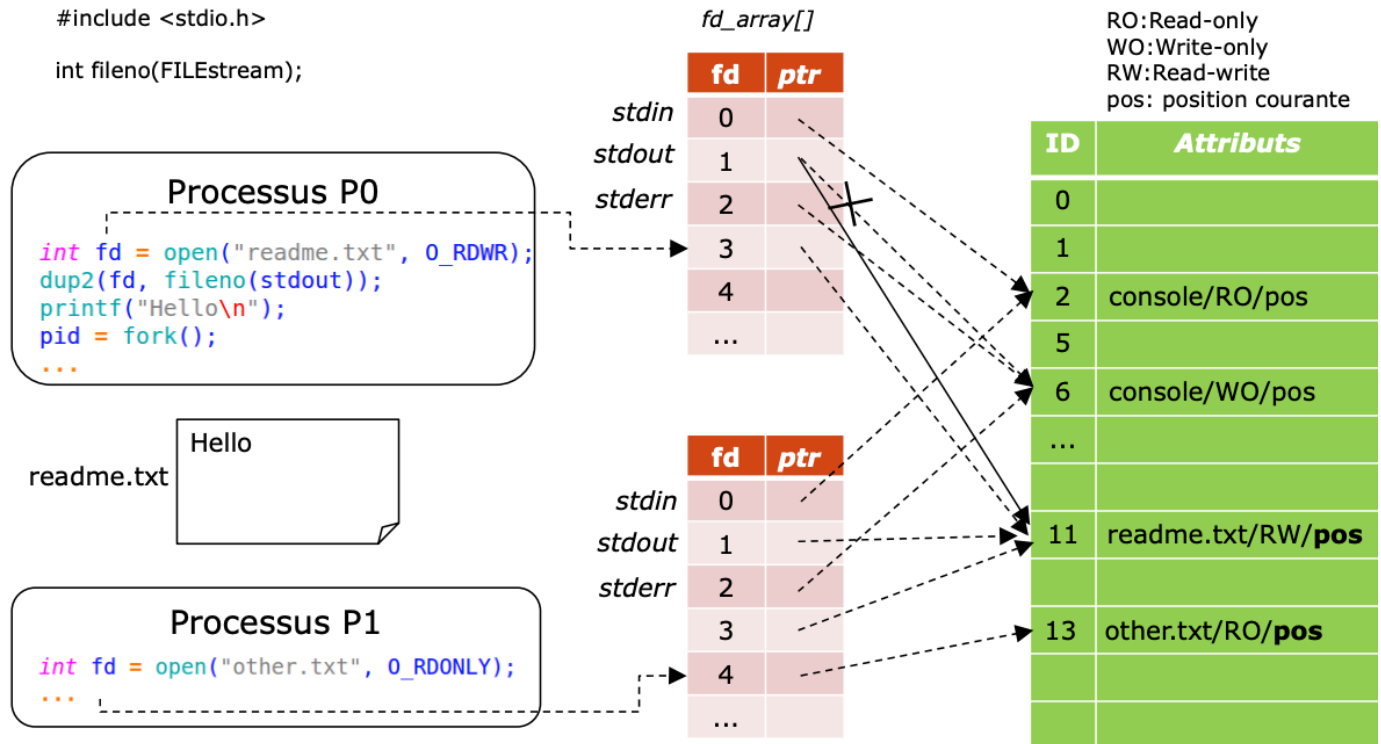
- 0 → stdin
- 1 → stdout
- 2 → stderr



- La table des fichiers ouverts stocke les informations nécessaires pour gérer les ressources entre les processus.
- L'ouverture d'un fichier dans différents processus conserve une position de lecture/écriture propre à chaque processus.
- La lecture dans un processus n'affecte pas celle d'un autre processus sur le même fichier.
- Lorsque plusieurs processus partagent la même console (comme depuis un shell), la position courante dans la console est commune.
- Dans ce cas, une seule entrée pour `stdout` suffit dans la table des fichiers ouverts.

1.4. Commande dup2

- L'appel système `dup2()` permet de rediriger un descripteur pour qu'il pointe vers une entrée de table utilisée par un autre descripteur.
- Exemple : rediriger `stdout` (descripteur no. 1) vers une entrée associée à un fichier, comme `readme.txt` (via le descripteur no. 3).
- Après cette redirection, toute écriture dans `stdout` sera enregistrée dans le fichier `readme.txt`.
- La redirection est spécifique au processus, mais elle est conservée si un appel à `fork()` est effectué.



- Tout comme dans Linux, la gestion des descripteurs de fichier est pris en charge dans une couche d'abstraction appelée VFS (Virtual File System).
- Chaque processus comporte sa propre table de descripteurs de fichier contenu dans le **PCB** du processus : c'est le tableau **fd_array** qui possède une référence vers la table des fichiers ouverts présente dans VFS (`open_fds`).

2. Sous-système IPC

- IPC signifie **Inter-Process Communication**
- Ensemble de mécanismes permettant la communication et la synchronisation entre processus

2.1. Propriétés des IPCs

- Les processus possèdent chacun leur propre espace d'adressage, les empêchant d'interférer entre eux.
- Contrairement aux processus, les threads d'un même processus partagent le même espace mémoire, facilitant leur communication et synchronisation.
- Pour permettre aux processus de partager des données ou de se synchroniser (ex. : copier-coller entre applications), le noyau implémente des mécanismes de communication inter-processus (IPC).
- Les IPCs sont gourmands en ressources CPU et mémoire, et font l'objet d'optimisations constantes.
- POSIX propose des spécifications variées pour la gestion des IPCs.
- Echanges de données entre les processus
 - Fichiers mappés
 - Segments de mémoire partagé
- Synchronisation entre les processus
 - Verrous, sémaphores, moniteurs, etc.
 - Signaux
- Echange et synchronisation entre les processus
 - Tubes (pipes)
 - Sockets réseau