

# Système d'Exploitation Unix

## TP

### Les appels système : gestion de processus (suite)

Tous les programmes devront être développés avec passage de leurs éventuels paramètres à la fonction `main (int argc, char *argv [])`. Les valeurs de retour des appels aux primitives devront être testées et les messages d'erreurs affichés avec `perror`. Les messages d'erreurs à destination de l'utilisateur se feront sur le fichier standard des erreurs `stderr`.

#### Question 1 Le recouvrement : les primitives `exec ( )`

Les primitives de la famille `exec()` permettent de charger en mémoire de nouveaux programmes binaires en vue de leur exécution.

Les primitives de la famille `exec()` se différencient par la manière dont les arguments sont transmis. Ces arguments sont transmis soit sous forme d'un tableau (famille `execv()`), soit sous forme de liste (famille `execl()`) selon que la primitive utilisée a un nombre fixe ou variable de paramètres.

- a) Ecrire un programme qui charge un nouveau programme binaire et dont les arguments sont transmis selon les deux modes précédents.

Les primitives de la famille `exec()` se différencient également par la manière dont le programme à charger est recherché dans le système de fichiers. Soit la recherche est relative au répertoire courant, soit elle l'est par rapport aux répertoires spécifiés via la variable `PATH`.

- b) Ecrire un programme qui charge un nouveau programme binaire qui est recherché selon les deux modes précédents.

Les primitives de la famille `exec()` se différencient enfin par l'environnement conservé par le processus après recouvrement. Soit l'environnement reste inchangé, soit un nouvel environnement est transmis en paramètre.

- c) Ecrire un programme qui charge un nouveau programme binaire et qui dans un premier cas conserve le même environnement et dans un second cas acquière un nouvel environnement.

Au cours d'un recouvrement, les caractéristiques suivantes ne sont pas conservées selon les conditions spécifiées :

- **propriétaire effectif** : si le `set-uid bit` est positionné sur le fichier exécutable chargé, le propriétaire de ce fichier devient propriétaire effectif du processus,
- **groupe effectif** : si le `set-gid bit` est positionné sur le fichier exécutable chargé, le groupe propriétaire de ce fichier devient groupe propriétaire effectif du processus,

- **descripteur de fichier ouvert** : si le bit `FD_CLOEXEC` d'un descripteur a été positionné à l'aide de la primitive `fcntl()`, ce descripteur est fermé après un recouvrement.

d) Ecrire des programmes qui vérifient les affirmations précédentes.

Au cours d'un recouvrement, les attributs suivants sont hérités par le nouveau programme :

- identifiant de processus (ID) et identifiant du processus parent (PID),
- identifiant utilisateur réel (UID) et identifiant de groupe réel (GID),
- répertoire de travail courant,
- masque de création de fichier,
- verrous sur les fichiers,
- masque des signaux (cf. partie suivante),
- signaux pendants (cf. partie suivante).

e) Ecrire des programmes qui vérifient les affirmations précédentes.

## Question 2 Le shell

Ecrire le programme d'un shell simplifié capable d'exécuter, en premier plan ou en arrière plan, n'importe quelle commande entrée par l'utilisateur avec un nombre arbitraire de paramètres.

Fonctions utiles:

- `fork`,
- `wait`,
- `exit`,
- `execvp`,
- `fgets`,
- `strtok`,
- `strchr`,
- `strlen`,
- `strcpy`,
- `malloc`.