

Système 1

TP 6 noté : les Dalton

Début : 9h30 - Fin : 11h30 (a priori)

Les frères Dalton ont été (encore) condamnés à 1342 ans de prison. Leur mère, Ma Dalton, les a rejoint. Au pénitencier, leur principale occupation est de casser des cailloux. Ma, la mère, refuse de tenir le marteau et se contente de trier les cailloux. Joe casse les rouges, William les gris, Jack les noirs et Averell mange les verts.

Objectif

Vous allez devoir faire un programme principal (Ma) qui crée quatre fils (Joe, Jack, William et Averell). Ma va ouvrir et lire un fichier contenant la liste des cailloux à casser (les cailloux sont identifiés par un caractère : **r** pour les rouges, **g** pour les gris, etc. cf le fichier). En fonction de la couleur du caillou, Ma le donnera au bon frère. Lorsqu'elle n'a plus de cailloux à distribuer, elle enverra un signal d'arrêt à ses enfants, puis les attendra et s'arrêtera.

Les quatre frères Dalton se contenteront de compter les cailloux qui leur sont passés. Lorsqu'ils reçoivent le signal d'arrêt, ils afficheront le nombre de cailloux cassés puis s'arrêteront.

Vous devrez rendre sur arche trois fichiers : `TP6_ma.c`, `TP6_freres.c` et `TP6_cailloux.txt`. Vous avez 2 heures.

Voici une proposition de découpage de ce TP en étape. Vous allez devoir faire deux fichiers : `TP6_ma.c` et `TP6_freres.c`. À chaque étape, vérifiez que tout marche. Vous n'êtes pas obligé de conserver les tests intermédiaires dans le rendu final, ni de suivre ce découpage.

1. `TP6_ma.c` ouvrira le fichier `TP6_cailloux.txt` et le lira octet par octet
2. `TP6_freres.c` affichera le nom du processus (contenu dans `argv[0]`).
3. Dans `TP6_ma.c`, Ma Dalton créera Joe avec un `fork` puis fera un `execl`. Le processus créé aura comme nom `Joe` (deuxième argument de `execl`).
4. Ma créera un tube et le transmettra à Joe. N'oubliez pas de fermer les bouts du tube nécessaires.
5. Via ce tube, Ma transmettra à Joe tous les cailloux rouge. Puis lorsqu'elle a finit de lire `TP6_cailloux.txt`, elle enverra le caractère `e` (comme end). Joe comptera les cailloux puis lorsqu'il reçoit `e`, il affiche le nombre de cailloux reçu puis s'arrête.
6. Ma va attendre Joe avec `wait` ou `waitpid` puis affichera `Fin`.
7. Vous pouvez désormais terminer le programme. Ma Dalton ouvre le fichier, crée ses quatre fils, leur donne les cailloux via un tube. Les frères travaillent en parallèle, affichent le nombre de cailloux cassés et se finissent. Ma Dalton attendra ses quatre fils puis affichera `Fin`.

À la fin, l'affichage pourra donner quelque chose comme ceci :

```
fst-dep-e-m01:TPs_Systeme1_2021 ngdr$ ./a.out
bonjour Joe
Joe a casse 3 cailloux...
bonjour Jack
Jack a casse 3 cailloux...
bonjour William
William a casse 1 cailloux...
bonjour Avrelle
Averell mange 2 cailloux...
Fin...
```