

Aucun document ou support autre que le sujet ou les copies d'examen n'est autorisé.
 (la copie ou les brouillons du voisin ne sont pas des supports autorisés).
 Éteignez impérativement vos mobiles.

1 Exercice

- Écrire une commande `rateau n` permettant de créer n processus de sorte que les n processus soient tous fils directs du processus exécutant la commande. Avant de terminer, le processus initial devra prendre soin d'attendre la terminaison de chacun d'entre eux.
- Écrire une commande `fil n` permettant de créer n processus de sorte que les n processus soient organisés sous la forme d'un fil généalogique, i.e. que le n -ième processus soit le père du $n + 1$ -ième processus. On veillera à ce que les processus se terminent proprement, aucun orphelin ne devra être obtenu et les zombies seront tous supprimés.
- Écrire une commande `arbre n` permettant de créer un arbre de processus de sorte que l'arbre soit de hauteur n , et que l'étage i de l'arbre contienne exactement i processus. On veillera aussi à ce que tout se termine proprement.

2 Exercice

- Écrire une commande `redir cmd` permettant d'obtenir le même résultat que la commande shell `cmd a b c d < f`
- Écrire une commande `tube cmd1 cmd2` permettant d'obtenir le même résultat que la commande shell `cmd1 < f | cmd2 >> g`

3 Exercice

On se propose de réaliser le crible d'Ératosthène qui permet d'obtenir les nombres premiers par élimination. En partant de la liste des entiers et en commençant par 2, on choisit le premier nombre qui reste, puis on élimine tous ses multiples, et l'on recommence :

- entrée : 2 3 4 5 6 7 8 9 10 11..., sortie : 3 5 7 9 11... nombre premier : 2
- entrée : 3 5 7 9 11..., sortie : 5 7 11..., nombre premier : 3
- entrée : 5 7 11..., sortie : 7 11..., nombre premier 5
- entrée : 7 11..., sortie : 11..., nombre premier 7
- ...

Il est demandé de construire une collection de processus chacun chargé d'éliminer les multiples d'un nombre donné de la façon suivante (on prendra soin de prendre en compte un maximum de cas d'erreurs, et de documenter le code) :

- Écrire une commande `elimine n` qui
 - lit sur son entrée standard une suite de nombres codés comme des entiers natifs (`int`),
 - écrit dans un fichier de nom `premiers` et en fin de celui-ci, et codé comme une chaîne de caractères le premier nombre reçu, et
 - produit sur sa sortie standard la suite des nombres reçus qui ne sont pas des multiples du premier reçu (nombres codés comme des entiers natifs).

Exemple, si la commande reçoit le codage natif des entiers 2 3 4 5 6 7 8, la chaîne "2\n" doit être ajoutée en fin du fichier `premiers` et en sortie doit être produite la suite 3 5 7.

Si l'argument de la commande est 1, elle doit se contenter d'écrire dans le fichier `premiers` le nombre 1 (sous la forme de chaîne), produire la suite des entiers natifs à partir de l'entier 2 et jusqu'à l'entier correspondant au symbole `INT_MAX` (défini dans le fichier d'entête `limits.h`) sur sa sortie standard, ne rien lire sur son entrée.

La commande `elimine 0` devra elle lire son entrée standard mais ne rien produire sur sa sortie et ne rien écrire dans le fichier `premiers`.

- Fournir une ligne de commande shell permettant d'obtenir la suite des 5 premiers nombres premiers dans le fichier `premiers`.

3. Écrire une commande `crible n` qui devra, à l'aide de la commande `elimine` produire dans le fichier `premiers` la suite des n premiers nombres premiers. Pour cela, la commande `crible` devra initier autant de processus que nécessaires tous communiquant par des tubes (anonymes) de sorte que la sortie standard du processus exécutant la commande `elimine n` devra correspondre à l'entrée standard du processus exécutant `elimine n + 1`. Le premier de la chaîne de processus devra exécuter simplement `elimine 1` et le dernier `elimine 0`. On veillera à ce que la terminaison des processus soit correcte (pas d'orphelin ni zombi).
4. Comment garantir, après exécution de la commande `crible n`, que le fichier `premiers` ne contienne que la liste des n premiers nombres premiers? Quels sont les problèmes possibles? Comment s'en prémunir? Qu'aurait-il fallu modifier dans le(s) code(s) précédent?
5. Modifier la commande `crible` de sorte que si l'on termine la commande par envoi d'un signal, l'ensemble se termine le plus proprement possible, c'est-à-dire, pas de processus zombi ni orphelin, le fichier `premiers` devra être nettoyé (soit vidé de son contenu, soit supprimé), etc.