

Exercice 5

Difficulté : 79 points-virgules

On désire réaliser un programme pour communiquer p valeurs positives à n processus fils en utilisant un fichier et des signaux :

`distrib n v1 ... vp`

La valeur v_i doit être transmise par le processus père au processus $v_i \bmod n$ en l'écrivant dans un fichier temporaire unique, utilisé par tous les processus, et en envoyant le signal `SIGUSR1` au processus fils concerné, qui doit alors l'afficher. Lorsque toutes les valeurs ont été transmises, le processus père doit envoyer le signal `SIGUSR2` aux processus fils pour qu'ils se terminent.

L'exemple ci-dessous montre notamment les affichages que vous devrez scrupuleusement générer :

```
> ./distrib 4 17 2 48
/tmp/distrib.W7VoRb          # chemin du fichier temporaire utilisé
fils 1 affiche 17             # affichage par le fils 1 (1 = 17 mod 4)
fils 2 affiche 2
fils 0 affiche 48
```

La description ci-dessus est incomplète et vous devrez peut-être utiliser d'autres signaux pour réaliser cet exercice.

Pour rédiger votre programme, il est impératif de respecter les contraintes suivantes :

- vous utiliserez la fonction de bibliothèque `mkstemp` pour obtenir le fichier temporaire; ce dernier doit être supprimé lorsque le programme se termine;
- vous ne devez utiliser que les primitives système (ou assimilées comme telles); vous pouvez toutefois utiliser les fonctions de bibliothèque pour les affichages, les manipulations de chaînes de caractères ou de mémoire ou bien sûr la génération du fichier temporaire;
- pour des raisons d'efficacité, vous ne ferez pas d'appels redondants à des fonctions lentes (primitives système ou autres);
- vous vérifierez soigneusement les débordements de tableau (vous pouvez notamment utiliser la fonction de bibliothèque `snprintf` pour contrôler la taille de chaînes complexes);
- votre programme doit retourner un code de retour nul (`exit(0)`) si tout s'est déroulé sans erreur ou un code de retour non nul (`exit(1)`) si une erreur a été rencontrée;
- si votre programme est appliqué avec un nombre d'arguments incorrect, il doit afficher le message : `"usage : distrib n v1 ... vp"`.
- vous apporterez un soin particulier à la mise en forme de façon à rendre un code lisible et commenté à bon escient. Référez-vous au document « Conseils pour réussir vos TP et projets » mis à votre disposition sur Moodle et, si besoin, utilisez l'utilitaire `clang-format` avec la configuration donnée dans ce document;
- votre programme doit compiler avec les options `-Wall -Wextra -Werror -pedantic` sur `gcc` version 9.4 minimum (la version disponible sur la machine `turing.u-strasbg.fr`). Alternativement, vous pouvez utiliser l'image Docker `pdagog/refc` (version de `gcc` 12.2) Les programmes qui ne compilent pas au moins sur `turing` avec ces spécifications **ne seront pas examinés**.

Un script de test est mis à votre disposition sur Moodle. Celui-ci exécute votre programme sur des jeux de tests qui serviront de base à l'évaluation de votre rendu. La commande suivante permet de lancer les tests : `sh test5.sh`.

Vous devrez rendre sur Moodle un *unique* fichier nommé `distrib.c`.

Cet exercice est **individuel**. On rappelle que la copie ou le plagiat sont sévèrement sanctionnés.