

# Principes des systèmes d'exploitation

## TP noté

DUT 2

10 janvier 2020

Vous pouvez vous mettre en binôme. Vous avez le droit d'utiliser votre machine, internet, votre cours, ... Vous devez déposer dans la zone de rendu sur elearning avant 17h20 le code source de votre programme au format NOM1\_PRENOM1\_NOM2\_PRENOM2.c.

Votre programme doit **IMPÉRATIVEMENT** être aéré, commenté, correctement indenté, concis (ne dupliquez pas du code), ne pas produire de messages d'alertes à la compilation, les noms des variables et des fonctions doivent être pertinents, ...

### Programme à réaliser

Votre programme doit prendre deux noms de fichiers (*fichier1* et *fichier2*) par paramètre et doit réaliser les 7 points suivants.

- 1) Créer deux processus fils nommés  $f_1$  et  $f_2$ .
- 2) Le père lit chaque seconde un caractère du fichier *fichier1*. Si c'est une lettre de l'alphabet (a-z ou A-Z) alors le père la transmet au fils  $f_1$ , si c'est un chiffre (0-9) alors il le transmet au fils  $f_2$ , sinon il l'écrit sur la sortie standard d'erreur.
- 3) Le fils  $f_1$  doit écrire sur la sortie standard les codes ASCII des lettres qu'il reçoit du père.
- 4) Le fils  $f_2$  doit écrire dans un fichier nommé *fichier2* les chiffres qu'il reçoit du père.
- 5) Votre programme doit s'arrêter lorsque que le fichier *fichier1* est entièrement traité. Veillez à la bonne fermeture des fichiers, processus, tubes (pipes), ...
- 6) La première fois que l'utilisateur envoie un signal d'interruption (Ctrl-C) seul le fils  $f_1$  est interrompu. Le fils  $f_2$  n'est interrompu qu'au deuxième signal d'interruption. Votre programme ne s'interrompt qu'au troisième signal.
- 7) Votre programme doit vérifier le bon fonctionnement de tous les appels systèmes. Si un appel système ne s'est pas correctement déroulé alors votre programme doit expliquer le problème sur la sortie standard d'erreur avant de s'arrêter.

Utilisez les fonctions *read*, *write*, *fork*, *pipe*, ... N'utilisez pas *printf*.