

TD n°7 – Bash, suite

1) Lancer-progs

NB : ce script est issu du TP7 qui n'a pas pu être terminé.

On voudrait programmer un script très utile dans un système d'exploitation. Ce script appelé `run-parts` dans le vrai système exécute chaque programme d'un certain dossier, en leur passant à chacun les mêmes paramètres.

D'abord, il faut créer un sous-dossier contenant quelques scripts ultra simples. Tapez ces commandes ou téléchargez le fichier

https://perso.univ-rennes1.fr/pierre.nerzic/SYS1A/data/td7_creerscripts.sh :

```
mkdir dossier
for nom in 06-s2.sh 01-script1.sh 45-balablabla.sh 13-essai.sh
do
    echo '#!/bin/bash' > dossier/$nom
    echo -n echo je suis le script $nom >> dossier/$nom
    echo ' et mes paramètres sont $@' >> dossier/$nom
    echo 'sleep $((RANDOM % 5 + 4))' >> dossier/$nom
    echo echo je suis $nom et je me termine >> dossier/$nom
    chmod u+x dossier/$nom
done
```

Comme vous le voyez, ça crée 4 scripts dans un dossier. Chacun, quand on le lance, affiche son nom et les paramètres qu'il reçoit, ensuite il attend entre 4 et 8 secondes puis affiche un dernier message avant de se terminer. Exemple à essayer ensuite : `dossier/01-script1.sh titi toto`

On remarque que les noms sont structurés en NN-reste. Ça permet de classer ces scripts : leur nom donne l'ordre dans lequel on doit les exécuter. On trouve cette technique de nommage pour avoir un ordre de prise en compte dans les dossiers `/etc/rc*.d`, dans `/etc/fonts/conf.d`, dans `/etc/X11/Xsession.d` (faites un `ls` sur ces dossiers).

On demande de faire un script appelé `lancer-progs` qui va lancer automatiquement tous ces scripts dans le bon ordre. NB : il peut y avoir autre chose que des scripts dans le dossier : des fichiers texte, des images... il ne faut lancer que ceux qui sont exécutables (script ou programme).

`lancer-progs dossier mot1 mot2` doit lancer `01-script1.sh`, puis `06-s2.sh` puis `13-essai.sh` puis `45-...` chacun avec les deux paramètres `mot1 mot2`.

Éléments : il faut faire une boucle sur tous les fichiers du dossier, pour chacun, tester s'il est exécutable : `test -x $nomfich`, si c'est le cas, alors le lancer en lui passant les paramètres qu'on a fourni à `lancer-progs`.

a) Lancement en arrière-plan

Amélioration : actuellement votre script lance successivement tous les programmes du dossier : il fonctionne de manière synchrone : le suivant ne démarre que lorsque le précédent s'est fini. On voudrait maintenant que si on fournit l'option `--jobs` à `lancer-progs`, il lance les programmes en arrière-plan. Du coup, l'ordre de lancement n'est pas important car tous vont démarrer presque en même temps. La commande interne `shift` vous servira sûrement.

lancer-progs --jobs dossier riri loulou doit lancer 01-script1.sh, puis 06-s2.sh puis 13-essai.sh puis 45-... chacun en arrière-plan, et tous avec les paramètres riri loulou.

b) Lancement dans l'ordre inverse

On souhaiterait maintenant qu'il y ait une option --inverse permettant de lancer les programmes dans l'ordre inverse. Au lieu de lancer 01-script1.sh en premier, il le lance en dernier.

C'est utile quand les scripts construisent ou défont quelque chose en fonction de leur paramètre.

```
lancer-progs dossier construire  
lancer-progs --inverse dossier detruire
```

On imagine que les scripts du dossier font un traitement différent selon leur premier paramètre. Par exemple, quand c'est construire, ils créent un fichier temporaire et quand c'est detruire, ils le suppriment.

Ce qui peut être utile, c'est l'option -r de la commande ls. Relisez aussi ce qui concerne les tableaux en bash.

2) Script NettoyerC

Écrire un script bash qui supprime les exécutables d'un dossier quand le source .c du même nom existe. L'idée est de gagner un peu de place sur le disque sans perdre pour autant le travail.

Voici ce que ça donne sur un exemple :

```
prompt% ls  
essai.c    essai      prog       tva.c      tva  
prompt% NettoyerC  
prompt% ls  
essai.c    prog       tva.c  
prompt%
```

Aucune indication n'est donnée, mais comme d'habitude, il faut essayer de voir ce que ça donne sur un fichier puis généraliser avec une boucle et enfin rajouter les tests garantissant la solidité du script.