

INF203 - Travaux pratiques, séance 3

Structures de contrôle en shell

Comme d'habitude, commencez par installer le répertoire TP3 à l'aide de `installeTP.sh`. Vous ferez ceci systématiquement au début de chacune des prochaines séances, ce ne sera plus indiqué dans les énoncés.

1 Instructions conditionnelles.

[TP3] Lisez le texte du programme `max2.sh`. [a] Comparez ce programme avec `max2.sh` du TP2. Quelles sont les différences de comportement des deux programmes? Quels tests effectuez-vous pour vérifier que ce programme est "correct" dans tous les cas de figure possibles? ■

Exercice complémentaire :

[b] Dans le programme `max2err.sh`, une erreur a été commise : laquelle? Cependant, ce programme fonctionne bien pour certaines configurations des données, lesquelles? En quoi ceci a-t-il un rapport avec la terminologie que nous utilisons : "interpréteur de commandes"? ■

Exécutez la commande `max2.sh` avec un nombre insuffisant d'arguments. Modifiez `max2.sh` pour afficher un message d'erreur dans ce cas de figure.

[c] Joignez le texte de `max2.sh` à votre compte-rendu. ■

Exercice complémentaire :

[INF203] Modifiez le fichier de commande `installeTP.sh` de manière à ce que la commande `installeTP.sh i`
— s'exécute normalement lorsque le répertoire `TPi` n'existe pas dans le répertoire courant
— affiche un message d'erreur dans le cas contraire.
Testez cette nouvelle version de `installeTP.sh`.

2 Boucles

[TP3] Lisez le texte du programme `description.sh`. Il indique, pour chaque élément du répertoire courant, s'il s'agit d'un fichier exécutable ou non, ou d'un répertoire.

Exercice complémentaire :

Modifiez ce programme pour qu'il crée, si nécessaire, un répertoire `Exec` dans le répertoire courant et y déplace les fichiers exécutables. Attention à ne pas déplacer le script lui-même!
[d] Testez cette nouvelle version et joignez le texte à votre compte-rendu. ■

2.1 Enchaînement de l'exécution de deux programmes : le retour

Reprenez le programme `instant_suivant.sh` du précédent TP (terminez-le si nécessaire; à défaut, utilisez celui fourni dans le répertoire TP3). Créez-en une copie nommée `instant_suivant10.sh` et modifiez ce programme pour qu'il répète 10 fois les opérations de lecture d'une heure au clavier et d'affichage de l'instant suivant.

Créez ensuite un script shell `des_heures.sh` qui affiche 10 fois de suite l'heure, à intervalles de 2 secondes. Consultez pour cela le manuel de la commande `sleep`, et pensez à utiliser une boucle... Testez vos deux programmes en fournissant en entrée de `instant_suivant10.sh` les sorties produites par `des_heures.sh`.
[e] Quelle(s) commande(s) utilisez-vous pour cela? ■

Exercice complémentaire :

Modifiez maintenant le programme `des_heures.sh` pour que le nombre d'itérations et le temps de pause entre deux affichages soient fournis en arguments de la ligne de commande.
[f] Testez cette nouvelle version et joignez le texte à votre compte rendu. ■
Essayez maintenant d'utiliser les sorties produites par cette nouvelle version de `des_heures.sh` comme entrées de `instant_suivant10.sh` avec pour valeur du nombre d'itérations, 10 puis 5 puis 15, par exemple..
[g] Quels comportements et/ou messages d'erreurs constatez-vous? ■

La commande de la semaine : diff

Pour savoir si les contenus de 2 fichiers sont identiques, on peut utiliser la commande `diff`. Elle donne 3 types de réponses :

- aucune réponse : les 2 fichiers sont identiques.
- la réponse est seulement que les 2 fichiers sont différents : l'un au moins n'est pas un fichier texte.
- la liste des différences entre les 2 fichiers : les 2 fichiers sont des fichiers texte.

En utilisant des copies des fichiers utilisés ou créés au cours de ce TP, faites en sorte d'obtenir ces 3 types de réponses avec la commande `diff`.

Autre commande de la semaine : basename

Lisez et exécutez `prefixe.sh`, et vérifiez que vous comprenez son fonctionnement.

3 Exercice de synthèse

Il s'agit d'écrire un script qui vérifie qu'un programme fournit les résultats attendus. Le programme à tester est `instant_suivant.sh` que vous avez récupéré en début de TP. Commencez par créer un répertoire `TEST_INSTANT_SUIVANT`, placez-vous dans ce répertoire et copiez-y `instant_suivant.sh`.

[`TEST_INSTANT_SUIVANT`] Supprimez tous les messages superflus de votre programme (par exemple "nouvelle heure :"). Assurez-vous que les entrées et les sorties soient réduites à :

- en entrée : une heure au format `HH:MM:SS`
- en sortie : l'instant à la seconde qui suit, au même format (sauf éventuellement les nombres inférieurs à 10 qui seront écrits avec un seul chiffre).

Votre programme doit avoir exactement le comportement suivant :

```
./instant_suivant.sh
14:47:25
14:47:26
```

Créez un certain nombre de fichiers d'entrée pour `instant_suivant.sh` : dans cette version, chaque fichier contient exactement une heure au format `HH:MM:SS`, et suffixez ces fichiers par `.entree`. Par exemple, le fichier `144725.entree` contient la chaîne `14:47:25`.

Pour chacun de ces fichiers, créez un fichier de même préfixe mais suffixé par `.sortie` contenant le résultat attendu du programme. Par exemple, le fichier `144725.sortie` contient `14:47:26`.

3.1 Test élémentaire

Exécutez les commandes :

```
./instant_suivant.sh < 144725.entree > 144725.ma_sortie
diff 144725.sortie 144725.ma_sortie
echo $?
```

Est-ce que le test est réussi ?

Déplacez le fichier *test_elem.sh* du répertoire [TP3] dans [TEST_INSTANT_SUIVANT].

Lisez, comprenez et exécutez ce programme.

3.2 Tests multiples

Et maintenant, écrivez un script *test_instant.sh* qui vérifie que votre programme *instant_suivant.sh* fournit bien les résultats attendus pour chacun des fichiers *.entree*. Pour cela, vous devez réutiliser un certain nombre d'éléments vus au cours des deux dernières semaines comme :

- le parcours dans une boucle *for* de tous les fichiers du répertoire suffixés par *.entree*
- l'extraction du préfixe de chacun des noms de fichiers
- ...

Testez ce programme, et utilisez-le pour vérifier que votre programme *instant_suivant.sh* est correct.

[h] Joignez le texte de *test_instant.sh* à votre compte rendu. ■