

À partir d'aujourd'hui tous les programmes shells que vous écrirez devront être placés dans le répertoire \$HOME/bin.

Transformations de la ligne de commandes

Exercice 1 : Protections

Dans cet exercice vous allez devoir utiliser la commande nommée `montrer-arguments` qui est disponible dans le répertoire `/home/public/m1101` et sur moodle. Cette commande permet de voir le nombre d'arguments qu'on lui passe ainsi que la valeur de ceux-ci. Elle n'est donc utile que pour essayer d'identifier les lignes de commande telles qu'elles sont réellement exécutées, c'est-à-dire après les transformations faites par le shell.

Q 1. Copiez le fichier `/home/public/m1101/montrer-arguments` dans votre répertoire courant.

Q 2. Donnez vous le droit d'exécution sur ce fichier.

Q 3. Écrivez des lignes de commande de façon à faire en sorte que la commande `./montrer-arguments` (qui sera donc le premier mot de chacune des lignes) affiche les résultats suivants :

1. nom de la commande \$0 = ./montrer-arguments
 nombre d'arguments \$# = 1
 argument numéro 1 \$1 = echo bonjour le monde

2. nom de la commande \$0 = ./montrer-arguments
 nombre d'arguments \$# = 4
 argument numéro 1 \$1 = echo
 argument numéro 2 \$2 = bonjour
 argument numéro 3 \$3 = le
 argument numéro 4 \$4 = monde

3. nom de la commande \$0 = ./montrer-arguments
 nombre d'arguments \$# = 3
 argument numéro 1 \$1 = echo
 argument numéro 2 \$2 = \t
 argument numéro 3 \$3 = plus loin

4. nom de la commande \$0 = ./montrer-arguments
 nombre d'arguments \$# = 2
 argument numéro 1 \$1 = echo
 argument numéro 2 \$2 = *

5. nom de la commande \$0 = ./montrer-arguments
 nombre d'arguments \$# = 3
 argument numéro 1 \$1 = echo
 argument numéro 2 \$2 = \t
 argument numéro 3 \$3 = *

6. nom de la commande \$0 = ./montrer-arguments
 nombre d'arguments \$# = 3
 argument numéro 1 \$1 = echo
 argument numéro 2 \$2 = sed
 argument numéro 3 \$3 = s/\./\t/g

7. nom de la commande \$0 = ./montrer-arguments
 nombre d'arguments \$# = 4
 argument numéro 1 \$1 = echo
 argument numéro 2 \$2 = 2
 argument numéro 3 \$3 = *
 argument numéro 4 \$4 = 3

- | | | |
|-----|--------------------|---------------------------|
| 8. | nom de la commande | \$0 = ./montrer-arguments |
| | nombre d'arguments | \$# = 4 |
| | argument numéro 1 | \$1 = expr |
| | argument numéro 2 | \$2 = 2 |
| | argument numéro 3 | \$3 = * |
| | argument numéro 4 | \$4 = 3 |
| | | |
| 9. | nom de la commande | \$0 = ./montrer-arguments |
| | nombre d'arguments | \$# = 1 |
| | argument numéro 1 | \$1 = B=1 |
| | | |
| 10. | nom de la commande | \$0 = ./montrer-arguments |
| | nombre d'arguments | \$# = 1 |
| | argument numéro 1 | \$1 = A=\${B} + 3 |
| | | |
| 11. | nom de la commande | \$0 = ./montrer-arguments |
| | nombre d'arguments | \$# = 3 |
| | argument numéro 1 | \$1 = A=\${B} |
| | argument numéro 2 | \$2 = * |
| | argument numéro 3 | \$3 = 3 |
| | | |
| 12. | nom de la commande | \$0 = ./montrer-arguments |
| | nombre d'arguments | \$# = 2 |
| | argument numéro 1 | \$1 = echo |
| | argument numéro 2 | \$2 = \${A} |
| | | |
| 13. | nom de la commande | \$0 = ./montrer-arguments |
| | nombre d'arguments | \$# = 2 |
| | argument numéro 1 | \$1 = expr |
| | argument numéro 2 | \$2 = \${A} |
| | | |
| 14. | nom de la commande | \$0 = ./montrer-arguments |
| | nombre d'arguments | \$# = 4 |
| | argument numéro 1 | \$1 = expr |
| | argument numéro 2 | \$2 = \${B} |
| | argument numéro 3 | \$3 = * |
| | argument numéro 4 | \$4 = 5 |

Exercice 2 : Jokers

Dans cet exercice vous ne devez pas utiliser la commande `cd`.

Les logins des étudiants du département sont créés à partir des 7 premières lettres de leur nom auxquelles on accole la première lettre de leur prénom. Un étudiant nommé Athanagor WURLITZER aurait donc comme login `wurlitza`.

Chaque étudiant du service informatique a son répertoire de travail dans un répertoire de `/home` dont le nom se termine par `etu`. Par exemple les étudiants du département informatique sont dans `/home/infoetu`, ceux du département génie biologique sont dans `/home/bioetu`, etc.

Q 1. Écrivez les lignes de commande, commençant par `echo`, qui permettent **d'afficher** les chemins absolus des répertoires principaux de :

1. tous les étudiants du département informatique;
2. tous ceux dont le nom et le prénom commencent par le caractère `b`;
3. tous ceux dont le login contient exactement 8 caractères. Il ne devra y avoir qu'un seul chemin par ligne (on rappelle que la séquence d'échappement `\n` correspond à un changement de ligne);
4. tous ceux dont le login fait exactement 8 caractères et dont le nom et le prénom commencent par le caractère `b`;
5. tous ceux dont le login contient au moins une fois le caractère `b`
6. tous ceux dont le login contient au moins deux fois le caractère `b`
7. tous ceux dont le login contient au moins un chiffre
8. tous ceux dont le login contient au moins une fois le caractère `t` et se termine par un chiffre

Exercice 3 : Substitutions

Q 1. Exécutez la commande `touch montrer-arguments`

Q 2. Après avoir lu attentivement la page du manuel de la commande `date`, donnez les lignes de commandes permettant :

- d’afficher la date du moment sous la forme `AAAA-mm-JJ-HH-MM-SS` où `AAAA` représente l’année courante, `mm` le mois courant, `JJ` le jour du mois, `HH` l’heure courante, `MM` la minute courante et `SS` la seconde courante ;
- de **renommer** (*i.e.* ajouter un nouveau nom) le fichier `montrer-arguments` en suffixant son nom initial par un point suivi de la date du jour formattée sous la forme `AAAA-mm-JJ-HH-MM-SS` ;
- de **renommer** le fichier `montrer-arguments` en suffixant son nom initial par un point suivi de la date de dernière modification du fichier formattée sous la forme `AAAA-mm-JJ-HH-MM-SS` (on rappelle que la commande `stat` permet d’obtenir toutes les caractéristiques d’un fichier).

Exercice 4 : Manipulations de variables : PATH

La variable `PATH` contient une liste de répertoires. Ils sont séparés les uns des autres par le caractère `:`.

Lors de l’exécution d’une commande le shell recherche les fichiers à exécuter uniquement dans les répertoires présents dans cette liste.

Q 1. Écrivez une ligne de commandes permettant d’afficher tous les répertoires contenues dans la variable `PATH`, **un par ligne**.

Q 2. Créez une variable `path_originel` en lui affectant le contenu de la variable `PATH`.

Q 3. Faites afficher le contenu de la variable `path_originel` puis celui de la variable `PATH` et vérifiez qu’ils sont bien identiques.

Q 4. Démarrez un nouveau terminal en exécutant la commande `x-terminal-emulator`.

Q 5. Dans ce nouveau terminal :

1. Videz le contenu de la variable `PATH`
2. Listez le contenu de votre répertoire
3. Restaurez le contenu de la variable `PATH` à sa valeur originelle en utilisant la variable `path_originel`
4. Fermez le terminal que vous venez d’ouvrir

Q 6. Ajoutez le répertoire `/home/public/m1101` en tête de la variable `PATH` puis exécutez la commande `ls`

Q 7. Ajoutez une ligne à la fin de votre fichier `$HOME/.bashrc` de façon à faire en sorte d’ajouter le répertoire `$HOME/bin` en tête de la variable `PATH`.

Exercice 5 : Quelques tests préliminaires

Les scripts que vous écrivez dans cet exercice ne doivent produire aucun affichage quelque soit les circonstances.

Q 1. Écrivez une commande, nommée `pareil`, qui ne réussit que si les deux paramètres qu’elles doit recevoir sont identiques.

Q 2. Étudiez la ligne suivante et expliquez ce qu’elle permet de faire si on l’inclut dans un script :

```
PATH=$(cd $(dirname $0) ; pwd):$PATH
```

Q 3. Écrivez une commande, nommée `est-vide`, qui ne réussit que si elle est appelée sans paramètre. Elle ne doit produire aucun affichage.

Q 4. Écrivez une commande, nommée `est-repertoire`, qui ne réussit que si le paramètre qu’elle doit recevoir est un répertoire. Elle ne doit produire aucun affichage.

Q 5. Écrivez une commande, nommée `est-executable`, qui ne réussit que si le paramètre qu’elle doit recevoir est un fichier (de n’importe quel type) dont le droit `x` est fixé pour son propriétaire. Elle ne doit produire aucun affichage.

Q 6. Écrivez une commande, nommée `est-executable-par-tout-le-monde`, qui ne réussit que si le paramètre qu’elle doit recevoir est un fichier (de n’importe quel type) dont le droit `x` est fixé pour toutes les catégories d’utilisateur. Elle ne doit produire aucun affichage.

Q 7. Écrivez une commande, nommée `est-nombre-entier`, qui ne réussit que si le paramètre qu’elle doit recevoir est un nombre entier. Elle ne doit produire aucun affichage.