path

- Objet : path
- Niveau requis :
  débutant, avisé
- Commentaires : *Configurer le PATH.*
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !
- Suivi :
  à-tester
  - Création par smolski le 26/03/2013
  - Testé par .... le ....
- Commentaires sur le forum : C'est ici[1)]

Préambule

Je veux installer[2)] un nouveau logiciel (un jeu par exemple) : asciijump. Je me logue donc en root[3)] pour faire un :

```
apt-get update && apt-get install asciijump
```

L'installation se passe bien, je regarde dans le menu `Applications` → `Jeux` et constate que ce nouveau jeu ne s'affiche pas.
Je décide donc de lancer mon jeu depuis la console :

```
asciijump
```

Problème, j'ai ce superbe message :

retour de la commande

```
bash : asciijump : command not found
```

Conclusion

Le répertoire contenant l'exécutable de mon jeu n'est pas renseigné dans le $PATH du compte courant (en l'occurrence, root).

Généralités

Qu'est-ce que le $PATH ?

Cette erreur arrive assez souvent, vous tapez une commande et vous avez un *command not foud*. C'est le cas pour les jeux en root par exemple. Pourquoi ?

Je vous invite à taper dans votre terminal :

```
echo $PATH
```

Vous obtenez une liste de répertoires. Par exemple chez moi :

retour de la commande

```
/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games
```

A quoi ça sert ?
C'est simple, quand vous tapez une commande du genre

```
frozen-bubble
```

Le terminal va parcourir tous les répertoires présents dans le **$PATH** dans l'ordre afin de voir si il ne trouve pas un exécutable du nom que vous avez appelé.

Ici, avec *frozen-bubble*, il va chercher ici :

```
/usr/local/bin/frozen-bubble
```

si il ne trouve pas, il va regarder dans

```
/usr/bin/frozen-bubble
```

etc...

Le dernier répertoire contenu dans mon $PATH est le répertoire /usr/games, il va donc regarder si il trouve :

```
/usr/games/frozen-bubble
```

/usr/games/ étant le répertoire par défaut des jeux, il va bien y trouver l'exécutable :

```
/usr/games/frozen-bubble
```

et me lancer mon jeu favori :)

Changer les dossiers par défaut

Le problème que j'avais était que je ne pouvais pas lancer mon jeu (asciijump) directement depuis le compte root, à moins bien sur de taper le chemin[4] complet de l'exécutable asciijump[5].

Je regarde donc ce que contient le $PATH root :

```
echo $PATH
```

J'obtiens :

retour de la commande

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11
```

Je me rends compte que le dossier */usr/games* n'est pas présent.
Voici comment l'ajouter :

Il y a deux solutions :

- Soit vous ajoutez un ou plusieurs répertoires pour un utilisateur donné
- Soit vous ajoutez un ou plusieurs répertoires pour tous les utilisateurs

Je vous recommande la première solution bien évidemment.

Lien utile

- https://debian-facile.org/utilisateurs:smolski:tutos:path

Une autre explication simplifiée offerte par bendia sur le forum :

1. https://debian-facile.org/viewtopic.php?pid=307336#p307336

Remarque

Il ne faut jamais mettre le répertoire courant dans la variable PATH, c'est une possibilité pour un programme malicieux d'exécuter un autre programme qui ne ferait pas ce que l'on souhaite.

Si tu mets **:.:** dans le PATH, les exécutables sont aussi cherchés dans le répertoire courant, quelque soit l'endroit où tu es. Le problème se pose surtout pour les programmes setuid ou setgid car si ils ont été mal faits, on peut exécuter un autre programme avec les droits root.

Exemple

Supposons que tu télécharges en tant qu'utilisateur un programme **malicieux** (un script qui rm -rf /n (Supprime toute la racine de ton système), par exemple), tu te trouves dans le répertoire de ce script qui porte le même nom qu'un utilitaire unix qui est utilisé par un programme setuid mal fichu.
Dans ce cas, si tu lances cette *commande_setuid*, la commande obtient les droits de root et peut donc s'engager sans que tu le saches.

La commande croit utiliser l'utilitaire unix et va exécuter le scripts mailicieux, et paf le / est effacé !
C'est pour cela, que dans les programmes **setuid** on vérifie systématiquement le PATH.

C'est une très vieille faille des système unix.

Sur les indications de **enikar***.*
Que ses neurones baignent à jamais dans le parfum des roses et des jasmins...

Un Utilisateur

Pour modifier le $PATH pour un utilisateur donné, éditez[6] le fichier de configuration de votre terminal pour le compte *utilisateur* :

```
nano /home/utilisateur/.bashrc
```

Et j'y ajoute cette ligne :

```
 PATH=$PATH:/usr/games
```

Pour root seulement

Pour ajouter le dossier /usr/**games** dans le *$PATH* du root.
J'édite en terminal root le fichier /root/.bashrc ainsi :

```
nano  /root/.bashrc
```

Ajouter la ligne :

```
 PATH=$PATH:/usr/games
```

Voila, désormais vous pouvez lancer votre jeu favoris depuis l'utilisateur root (*Je sais, aucun intérêt pour un jeu,mais c'est pour donner un exemple quoi*

Tous les Utilisateurs

Pour modifier le $PATH pour tous les utilisateurs, ajoutez la même ligne dans le fichier de configuration général de votre terminal qui doit se trouver ici : /etc/bash.bashrc.

```
nano  etc/bash.bashrc
```

Idem, ajouter :

```
 PATH=$PATH:/usr/games
```

Lien

- http://www.generation-linux.fr/index.php?post/2008/10/15/Changer-les-dossiers-par-defaut-dans-le-PATH

Merci à **Benjamin** *à qui je dédie les bisous de multiples générations d'utilisateur Linux pour la clarté de son exposé.* 😉

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

aptitude

3)

terminal

4)

repertoires

5)

/usr/games/asciijump

6)

Editeur nano