TD n°2 : Linux/Unix, la suite…

Programmation en langage structuré
L3 - EEA

Résumé

Jusqu'ici, nous avons étudié quelques commandes de base Unix. Dans ce TD nous allons nous attacher à des commandes plus avancées donnant une meilleure idée de l'intérêt d'un shell par rapport à un explorateur de fichiers graphiques.

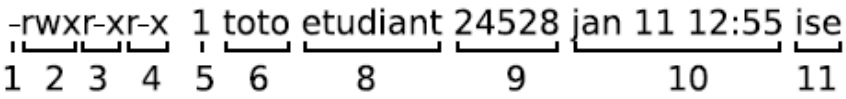
1. Le système de fichiers

1.1. Les droits

Dans votre répertoire home, créez un nouveau dossier _TP2_. Allez dans le répertoire (&home_prof)/ et essayez de créer à nouveau un dossier _TP2_ dans ce répertoire. Que se passe t'il ? Retournez dans votre répertoire home et tapez

% ls –l

Essayez de décrire les différents champs de chaque ligne tels que numérotés dans l'exemple ci dessous :



Retournez dans (&home_prof)/ et vérifiez vos droits en lecture sur le dossier TP2 dans ./progstruct/. Si vous disposez des droits nécessaires, copiez son contenu dans le répertoire TP2 créé en début de TD dans votre répertoire home.

Ouvrez un éditeur de texte, tapez les lignes ci dessous :

%clear
%ls

puis enregistrez ce fichier dans _/TP2/cls.

Placez-vous dans_/TP2/ et tapez

% cls

Que se passe t il ? Donnez au fichier cls les droits en écriture (avec chmod) puis recommencez. Que se passe t il ? Tapez maintenant ./cls. Que se passe t il ? Comment expliquer cette différence ?

1.2. Les fichiers cachés

Placez-vous dans votre répertoire home et tapez

% ls
% ls –a

Comment expliquer la différence ?

Quel est la particularité des fichiers cachés ?

1.3. Les wildcards

Placez-vous dans _/TP2/ et listez tous les fichiers PDF contenus en utilisant *ls* et un wildcard.

1.4. Les alias

Créez un alias "*la*" pour *ls -a* puis testez-le. Ouvrez un nouveau terminal et testez à nouveau la commande "*la*". Que se passe t il ? Expliquez...

1.5. Quelques mots concernant le shell

Tapez % *chsh -l* , commentez le résultat. Avec un éditeur de texte, modifiez le fichier *".bashrc*" qui se trouve dans votre répertoire *home* et ajoutez l'alias *"la"* ainsi qu'un alias *"ll"* pour *ls –al* (s'ils n'existent pas déjà) ainsi que la ligne *echo Bienvenue dans le terminal de votre_nom* . Fermez le terminal et en ouvrir un nouveau. Commentez...

1.6. Les liens

Créez un lien sur un des fichiers se trouvant dans votre répertoire TP2 en utilisant la commande *ln*. Vous appellerez le lien "lien_nom du fichier pointé". Listez en détail les fichiers contenus dans le répertoire, que remarquez-vous ? Vérifiez que le lien fonctionne en l'ouvrant par son nom originel ou par le lien créé. Supprimez maintenant le fichier originel, puis essayez à nouveau de l'ouvrir en appelant le lien. D'après cette expérience que concluez-vous sur la nature d'un lien ?
Renouvelez l'opération en utilisant cette fois ln -s (lien symbolique), conclure sur la différence entre un lien "physique" et un lien symbolique...

2. Les fichiers d'E/S

2.1. Edition de fichiers textes

Affichez le contenu du fichier "sujetTP2.tex" dans le terminal en utilisant la commande *cat*. Quel est l'inconvénient de cet affichage ? Recommencez en utilisant cette fois la commande *more* puis *less*. Commentez les différences notables entre ces trois méthodes.
Tapez maintenant % *grep* section sujetTP2.tex. Expliquez...

2.2. Les redirections d'E/S

Tapez la suite de commande suivante :

```
ls -l > list1
ls -l
more list1
ls -l >> list2
ls -l
more list2
ls -l > list1
ls -l >> list2
more list1
more list2
```

Qu'en concluez-vous sur la différence entre ">" et ">>" ? Tapez maintenant la suite de commandes suivantes :

```
ls -l
ls -l | grep pdf
```

Qu'en conclure sur la fonction du "|" ?

3. Les process

3.1. Arrière plan/premier plan

Lancez le programme "*xclock*" depuis le terminal (wahou une jolie horloge !). Essayez maintenant de lancer un éditeur de texte depuis ce même terminal sans arrêter *xclock*... Quittez *xclock* et relancez le en tapant

```
% xclock &
```

Expliquez... Laisser tourner l'horloge.

3.2. Enchainement de commandes

A la question 1.1, un fichier cls a été créé. Il permet d'exécuter la commande *clear* suivie de la commande *ls*. Il est possible de le faire en une seule ligne :

```
% clear ;ls
```

Essayez et concluez sur le " ;"... L'avantage ? Tapez à présent la commande

```
% alias cls 'clear ;ls'
```

Si vous jugez la commande cls pratique ajoutez-la à votre fichier ".bashrc".

3.3. En cas de plantage

Il peut arriver qu'un programme plante, il est donc important de savoir comment s'en sortir. L'horloge *xclock* lancée à la question 3.1 tourne toujours en tâche de fond, si ce n'est pas le cas, relancez-la. Lancez maintenant une seconde horloge *xclock* mais cette fois-ci en premier plan. Admettons que cette dernière ait planté (même si ce n'est pas le cas, cela ne fait pas de différence) tapez [ctrl]+[c] , que constatez-vous ?

Admettons maintenant que l'horloge qui tourne en tâche de fond a planté, réessayer la même méthode. Que se passe t il ?

Tapez maintenant

```
% ps -u votre_login
```

Cela permet d'afficher la liste de tous les process qui tournent sous votre session (y compris la fenêtre firefox qui n'a rien à faire là).

A côté du nom du process, se trouve un numéro : le PID de ce process. C'est la référence par laquelle le système d'exploitation connait ce process.

Pour tuer un process (par exemple parce qu'il a planté) on tape la commande

```
% kill -9 PID_du_process_a_tuer.
```

Utilisez les commandes *ps* et *kill* pour tuer la première horloge

Relancez l'horloge en tâche de fond et tapez la commande

```
% xkill
```

le pointeur de la souris change, il est alors possible de tuer le process associé à une fenêtre en cliquant sur cette fenêtre (il s'agit de la version graphique de kill -9).