

TP1 Linux

Dimitri Galayko

1. Démarrer l'ordinateur. Dans le menu d'amorçage, choisissez Linux.
2. Loggez vous (ouvrez la session).

1 Découverte de l'environnement graphique

1. Cliquez sur le bureau avec le bouton droit de la souris. Explorez toutes les fonctions proposées par le menu.
2. Créer sur le Bureau une icône permettant de lancer Mozilla.
3. Ouvrez une fenêtre de navigateur internet de votre choix et affichez la page google.fr
4. Ouvrez l'application OpenOffice Writer (l'équivalent du Word de Microsoft). Créez un fichier texte et enregistrez le sur bureau.
5. Ouvrez les applications équivalentes à Power Point et Calculator de Microsoft dans l'environnement de OpenOffice.
6. Explorez le menu "Applications".

2 Ligne de commande du terminal

1. Ouvrez un terminal. Avec la commande *pwd* affichez votre emplacement dans l'hierarchie du système de fichier.
2. Avec la commande *ls* affichez la liste du répertoire.
3. Tapez *ls -a*. Qu'est-ce qui change par rapport à la commande *ls*?
4. Tapez *ls -al*. (Avec l'aide du moniteur), vous devez comprendre la signification de différentes champs. Notamment, vous devez pouvoir donner la nature du fichier (fichier ou répertoire), sa taille en octets, la date de création ou modification, le propriétaire, le groupe, et les permissions. Vous devez comprendre également le significations des répertoires "." et "..".

5. Allez dans le répertoire `/usr/bin`. Affichez son contenu avec la commande `ls -al`. Vous remarquerez qu'il est impossible de faire tenir la liste de tous les fichiers/répertoires sur un écran. Utilisez la commande `ls -al | less`. Commentez.

6. Pour revenir à votre répertoire *home*, tapez `cd` sans arguments.

7. Quelques astuces de navigation dans le système de fichiers : le répertoire `~` désigne toujours votre répertoire d'accueil (*home*). Pour tester, allez d'abord ailleurs, par exemple, dans le répertoire racine `/`, ensuite tapez `cd ~`. Vérifiez votre emplacement par la commande `pwd`.

De même, vous pouvez facilement indiquer un répertoire d'accueil d'un autre utilisateur en mettant la tilde devant son login : `cd ~galayko`. Dans le cas où l'utilisateur a ouvert son compte en lecture, vous vous retrouvez dans son répertoire (faites `pwd` pour vous en assurer). Sinon, vous aurez un message du genre :

```
-bash: cd: /home/profs/galayko: Permission denied
```

8. La fenêtre du terminal garde en mémoire les anciennes commandes. Pour les rappeler, utiliser les flèches "haut" et "bas" du clavier.

9. Le terminal offre un système de suggestion puissant. Notamment, l'aide de deux types est offerte.

a) suggestion de commandes executables : en appuyant sur la touche "tabulation", le terminal vous proposera les commandes disponibles dans votre environnement (essayer !). Il y en aura beaucoup (chez moi c'était 2300!); il vous demandera si vous souhaitez toutes les afficher (la réponse raisonnable, c'est non).

Par contre, c'est très utile si vous vous souvenez les premières lettres du nom de commande. En tapant `gr+tabulation`, le terminal affichera toutes les commandes commençant par "gr". Vous pouvez choisir et spécifier la troisième lettre (taper `e+tabulation`): vous aurez l'unique suggestion, la commande `grep` (que l'on étudiera plus tard). C'est surtout pratique lorsque le nom de la commande est long.

b) De même, une fois la commande entrée, le plus souvent on passe des fichiers en arguments. Le terminal offre le même système pour la suggestion du nom de fichier. Par exemple : `grep+espace+tabulation` donnera la liste de fichiers se trouvant dans le répertoire courant. En spécifiant les premières lettre du nom de fichier souhaité, vous pouvez économiser la saisie du nom complet.

La même chose pour les noms de fichiers avec des chemins complets :
`grep+espace+/+tabulation`

vous donne le contenu du répertoire /. Vous pouvez ensuite spécifier un répertoire, puis un sous-répertoire, et ainsi de suite, jusqu'au nom de fichier.

En faisant cette opération, affichez le contenu du fichier /home/profs/-galayko/toto.

10. Dans votre répertoire *home*, allez dans le répertoire Desktop, affichez son contenu et assurez vous que le fichier Open Office Writer que vous avez créé est bien là.

11. Créez un répertoire toto1. Assurez vous qu'une icône correspondante a apparu sur le Bureau.

12. Copiez votre fichier créé sous Open Office dans le répertoire toto1.

13. Dans le répertoire toto1, créer un répertoire "toto2" et déplacez le fichier crée de "toto1" vers "toto2". / 14. En plus de traditionnels ctrl-C et ctrl-V pour faire du copier-coller, Linux offre un moyen de copier directement le texte sélectionné. Il suffit de positionner le curseur de la souris sur l'écran de destination et d'appuyer sur le bouton de milieu de la souris. Essayez de le faire.

3 Gestion de processus élémentaire

Linux offre un moyen très souple de gestion de processus, i.e., de programmes qui tournent en mémoire. L'utilisateur peut à tout moment "tuer" un process/programme - qui a planté ou dont il n'a plus besoin. Ceci distingue avantagement Linux de Windows.

Pour afficher la liste de processus, utiliser la commande

```
ps -ef
```

Vous avez la liste avec, de gauche à droite : le nom de la personne qui a lancé le process, le numéro du process, ensuite, d'autres informations systèmes, et en deriner, le nom du programme/process.

Exercice : lancer gimp (lecteur de fichiers graphiques) :

```
gimp
```

Affichez la liste de processus et trouvez le numéro du process correspondant à ce programme.

Pour tuer un process, il faut utiliser la commande

```
kill -9 <numéro de process>.
```

On ne peut tuer que les processus dont on est propriétaire (heureusement...).

On peut également tuer tous les processus dont vous êtes propriétaire en tapant ”-1” à la place du numéro. Vous serez retourné à la fenêtre de login : cette opération est équivalente à une fermeture de la session.

4 Systeme Linux. Utilisation avancée du terminal

Dans cette partie vous allez découvrir quelques astuces qui permettent d’accélérer votre travail sous terminal – et qui font la différence avec les systèmes d’interface graphique.

1. Périphériques de sortie.

Rappelons que sous Linux (Unix...) tout est fichier. Plus précisément : toute périphérique d’entrée-sortie est traitée (vue, représentée) comme un fichier.

Exemple : CDRom, clé USB... L’usage veut que ces dispositifs soient ”montés” dans le répertoire `/media/`.

Allez dans ce répertoire (commande `cd`) et affichez son contenu. Si vous avez une clé USB, connectez la et observez ce qui se passe. Normalement Linux détectera automatiquement l’insertion du dispositif, et créera un nouveau répertoire dans le répertoire `/media`, dont le nom dépendra du nom de la clé USB, et qui contiendra tous les fichiers se trouvant sur la clé.

Après l’insertion de la clé, reaffichez le contenu du répertoire `/media/<nom_de_la_clé>`.

Plus généralement, l’information sur l’intégration de dispositifs de stockage physiques dans le système de fichier virtuel se trouve dans le fichier `/etc/fstab`.

2. Redirection des entrées et des sorties.

De même que les dispositifs de stockage, un terminal est également un fichier : le système y envoie des données (le texte, les messages...) ou lit une information entrée au clavier par l’utilisateur, comme s’il s’agissait d’un fichier sur le disque. Le fichier, ou le flux (terme propre aux systèmes de famille Unix) qui correspond à un terminal s’appelle ”entrée-sortie standard”.

Ainsi, lorsque un programme génère un message destiné à l’utilisateur (par exemple, ”Entrez le nom du fichier”), il écrit (envoie) le texte du message dans le fichier (flux) de sortie standard.

Linux permet d’associer à la sortie standard n’importe quel fichier. L’opération qui permet de le faire s’appelle ”redirection d’entrée-sortie”. Cela se fait à

l'aide de l'opérateur désigné par les symboles ">" et "<".

Exemple.

Afin d'enregistrer le contenu d'un répertoire dans un fichier nommé, par exemple, "toto", plutôt que de l'afficher sur un écran :

```
ls > toto.
```

Rien n'est affiché sur le terminal, mais si l'on consulte le contenu du fichier toto nouvellement créé, on verra la liste qu'aurait fait afficher la simple commande ls.

Si on refait la commande `ls > toto`, l'ancien fichier toto sera écrasé et créé de nouveau. Cependant, très souvent on souhaite juste ajouter des informations aux données existant dans le fichier. Pour cela il suffit de faire

```
ls >> toto.
```

En faisant cette opération plusieurs fois, vous remarquerez que le fichier toto "gonfle" (vous pourrez le voir en consultant la taille de ce fichier avec la commande `ls -l toto`).

Exercices.

Dans le répertoire `/usr/bin`:

- Trouvez des fichiers dont le nom commencent par "em", finissent par "s".

- Définissez un filtre qui spécifiera en plus, que dans le nom, entre "em" et "s", il y a une lettre "a".

4.1 Caractères génériques : *, ? et []

Linux offre plusieurs outils puissants pour traiter des chaînes de textes.

Par exemple, on souhaite afficher tous les fichiers du répertoire commençant par un préfixe "pr". Ceci se fait à l'aide de la commande :

```
ls pr*
```

En effet, l'étoile "*" signifie "n'importe quelle chaîne de caractères, y compris une chaîne vide".

Par exemple, si on souhaite faire afficher tous les fichiers commençant par préfixe toto, on tapera:

```
ls toto*
```

Il existent d'autres caractères spéciales faisant partie des expressions régulières. Ainsi, le caractère "." signifie "n'importe quel caractère unique", une expression [liste de caractères] signifie "n'importe quel caractère de la liste", etc...

Pour pouvoir tester le fonctionnement des expressions régulières, introduisons la commande `grep`. Elle est utilisé dans le contexte suivant:

grep <expression régulière> <nom de fichier>.

Son rôle est d'afficher sur le terminal toutes les lignes du fichier *<nom de fichier>* contenant une chaîne de caractère correspondant à l'expression spécifiée.

Ainsi, la commande *grep * toto* affichera toutes les lignes du fichier *toto*.

La commande *grep . toto* affichera toutes les lignes contenant au moins un caractère.

La commande *grep .aaa toto* affichera toutes les lignes contenant la chaîne "aaa" suivi par n'importe quel caractère (autrement dit, les lignes contenant "aaa" et ne commençant par "aaa").

La commande *grep motif* affichera toutes les lignes commençant par *motif*.

La commande *grep motif\$* affichera toutes les lignes finissant par *motif*.

Dans cette introduction, vous n'avez découvert qu'une petite partie de la puissance des expressions régulières.

4.2 Empilement de commandes (pipelines)

C'est un outil extrêmement puissant de Linux, permettant de passer le résultat d'une commande à l'entrée de l'autre.

Quelques exemples.

La commande *less* affiche le contenu d'un fichier page par page.

less toto

La commande *ls* affiche le contenu du répertoire courant, mais sur plusieurs écrans, ce qui n'est pas commode (les noms qui viennent au début de la liste ne sont pas lisibles). On peut associer les deux commandes : *ls* génère la liste de noms, et cette liste est retransmise à la commande *less*, qui l'affiche page par page :

ls | less

La commande *grep* est un très puissant filtre pour un fichier de texte. Elle est utilisée d'une manière suivante :

grep <expression régulière> <nom de fichier>.

Cette commande affichera uniquement les lignes de fichier qui contiennent une chaîne de caractère correspondant à l'expression régulière spécifiée.

Exemple: récupérez le fichier *rc* (un fichier texte se trouvant dans le répertoire *~galayko*) et tapez :

grep run rc

Vous verrez sur l'écran uniquement des lignes contenant "run".

La commande *grep* peut être utilisée avec d'autres commandes pour sélectionner une information. Par exemple, on peut demander une liste détaillée du contenu d'un répertoire, et demander d'afficher tous les fichiers appartenant à un utilisateur :

```
ls -l | grep galayko
```

De même, il est possible de connaître quels sont les processus qui vous appartiennent :

```
ps -aux | grep <votre login>.
```

Cette empilement de commandes peut être multiple. Ainsi, si de toutes les processus qui vous appartiennent vous souhaitez voir ceux qui concernent le programme gimp, vous pouvez juste faire :

```
ps -aux | grep <votre login> | grep gimp.
```

En effet, la sortie de la commande *ps -aux | grep <votre login>* sera l'entrée de la commande *grep gimp*.

5 Travail personnel

1) Dans le répertoire personnel de l'utilisateur *galayko*, trouvez le répertoire *documents*. En explorant le contenu de ce répertoire, dessinez sur une feuille de papier son arborescence.

2) dans le répertoire */usr/include/* trouvez tous les fichiers avec les noms commençant par *key*. Vous utiliserez la commande *find* .