NSY103
TP 2 – Introduction à la ligne de commande sous GNU/Linux (partie 2)

2016

Les TPs seront réalisés sous GNU/Linux.

Pour commencer, démarrez une machine sous GNU/Linux et connectez-vous en utilisant utilisant les login/mot de passe génériques: `licencep` / `7002n*`

Partie 1 : Expansion de noms de Fichiers

Le shell permet de compléter automatiquement le nom de fichier à partir d'un motif. Cela permet de faire référence à des fichiers sans en connaître le nombre à priori ni même le nom exacte.

Par exemple, un répertoire peut contenir les fichiers suivants :

```
$ ls
prog1 prog1.c prog2 prog2.c prog2 prog2.c prog3 prog3.c proga proga.c
```

Si on souhaite lister uniquement les fichiers « `.c` », on peut exécuter la commande suivante :

```
$ ls *.c
prog1.c prog2.c prog2.c prog3.c proga.c
```

De même, on pourrait vouloir lister uniquement les fichiers ayant un chiffre dans leur nom.

Motif	Description
`*`	Indique la présence de zéro ou plusieurs caractères
`?`	Indique la présence d'exactement un caractère
`[...]`	Permet de spécifier la présence d'exactement un caractère parmi un ensemble de caractères donnés explicitement
`[.-.]`	Permet de spécifier la présence d'exactement un caractère parmi un ensemble de caractères définit par un interval (l'ensemble des lettres minuscules : [a-z])

Pour rechercher les fichiers commençant par `prog` et contenant le chiffre 2 et la lettre a, on procède ainsi :

```
$ ls prog[2a]*
prog2 prog2.c proga proga.c
```

Exercices

1. Donnez la commande permettant d'afficher les fichiers commençant par `prog` et contenant un chiffre ;
2. Donnez la commande permettant d'afficher les fichiers commençant par `prog` et contenant une lettre majuscule ou minuscule.

Partie 2 : Nom de Fichier et Caractères Spéciaux

On liste maintenant un autre répertoire qui contient les fichiers suivants :

```
$ ls
prog* prog␣1 prog␣1.c prog␣2 prog␣2.c prog␣2 prog␣2.c prog␣3 prog␣3.c
prog␣a prog␣a.c prog*.c
```

Remarque. *Notez la présence d'espace dans les noms de fichiers. Les espaces (␣) sont mis en évidence pour le TP. Ce ne serait pas le cas dans un terminal).*

Exercices

1. Listez les fichiers « `prog 1` » et « `proc 1.c` » ;
2. Que donne la commande « **`ls prog 1*`** » ?
3. Listez les fichiers contenant le caractère `*` ;
4. Ces commandes produisent-elles le résultat souhaité ?

Protection Contre l'Expantion

Afin de « protéger » certain caractère de l'expansion, on peut les faires précéder du caractère \.

```
$ ls prog\*
prog*
$ ls prog\␣1
prog␣1
```

Une autre possibilité est d'encadrer la chaîne de caractère avec des doubles quotes :

```
$ ls "prog␣1"
prog␣1
```

Exercices

À l'aide d'une seul commande :

1. Listez les fichiers « `prog 1` » et « `proc 1.c` » ;
2. Listez les fichiers commençant par « `prog` » et contenant un espace ;
3. Listez les fichiers contenant le caractère `*` ;

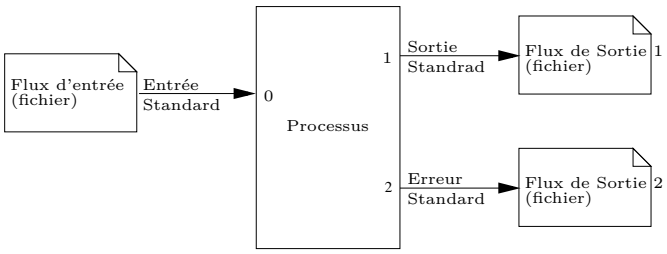

FIGURE 1 – Entrée / Sortie Standards d'un programme

Partie 3 : Entrée / Sortie Standards

a) Introduction

À chaque processus sont associées une entrée, une sortie et une sortie d'erreur dites standards. Par défaut, les processus lisent et écrivent sur ces entrées/sorties.

Par exemple, la commande **ls** écrit sur sa sortie standard la liste des fichiers contenus dans le répertoire courant :

```
$ ls

Archives              Documents             public_html
bin                   Downloads             Téléchargements
bonjour.txt           mbox                  test.txt
Bureau                Music                 tmp
Desktop               Pictures              Videos
```

La commande **cat** permet d'afficher le contenu d'un fichier sur sa sortie standard :

```
$ cat bonjour.txt
Bonjour le monde !
```

La commande **echo** permet d'afficher une chaîne de caratère sur sa sortie standard :

```
$ echo 'Bonjour le monde !'
Bonjour le monde !
```

Par défaut, les entrée et sorties standards d'un programme lancé en ligne de commande est le terminal dans lequel le programme a été lancé. Ainsi, pour les commandes précédentes, la sortie standard était le terminal et c'est sur celui-ci qu'on a pu lire le résultat des commandes.

b) Redirection des Entrée / Sorties

On peut souhaiter rediriger les entrée / sorties d'un programme, notamment pour sauvegarder son résultat.

Pour enregistrer dans un fichier la liste des fichiers contenus dans le répertoire courant, on utilise l'opérateur de redirection '**>**' pour rediriger la sortie standard de la commande **ls** vers un fichier :

```
$ ls > liste-fichiers.txt
$ cat liste-fichiers.txt
Archives     Documents  public_html
bin          Downloads  Téléchargements
bonjour.txt  mbox       test.txt
Bureau       Music      tmp
Desktop      Pictures   Videos
```

Le shell nous fournit les opérateurs de redirection suivant :

Opérateur	Description
`< nom_fichier`	redirection de l'entrée standard depuis ce fichier
`> nom_fichier`	redirection de la sortie standard vers ce fichier. Si le fichier existe, son contenu est écrasé
`» nom_fichier`	redirection de la sortie standard vers ce fichier. Les nouvelles données seront enregistrées à la fin du fichier
`2>` ou `2» nom_fichier`	indique la redirection de la sortie d'erreur standard vers ce fichier. Les données contenues dans ce fichier seront soit écrasées (`2>`), soit conservées (`2»`)
`&> nom_fichier`	indique la redirection de la sortie standard et de la sortie d'erreur standard vers ce fichier

Table 1 – Opérateurs de redirection

Exercices

1. Que font les commandes suivantes :
   (a) `$ cat f1`
   (b) `$ cat f1 > s1`
   (c) `$ cat f1 f2`
   (d) `$ cat f1 f2 > s2`
2. Quelle est la différence entre les deux séquences de commande suivantes :
   (a) `$ cat f1 f2 > s2`
   (b) `$ cat f1 > s2`
   `$ cat f2 » s2`
3. Donnez la commande permettant d'enregistrer la liste du contenu du répertoire `~/rep1` dans le fichier `~/liste_rep1.txt` ;

Partie 4 : Commandes avancées

a) Chaînage de commandes

Il peut être intéressant de passer la sortie d'une commande à une autre commande. Par exemple, essayez la commande suivante :

```
$ ls -l /usr/bin
```

La sortie de cette commande est très longue et s'affiche sur plusieurs pages du terminal. Il serait utile de l'afficher page par page grâce, par exemple, au pager **less**. Pour cela, on utilise l'opérateur '|' qui permet de rediriger la sortie standard d'un programme vers l'entrée standard d'un autre :

```
$ ls -l /usr/bin | less
```

b) Commandes de manipulation de fichiers text

Commande	Arguments	Description
head	-n x	affiche uniquement les x premières lignes
tail	-n x	affiche uniquement les x dernières lignes
tr	'c' 'b'	remplace toutes les occurrences du caractères 'b' par le caractères 'c'
tr	-s 'c'	remplace les occurrences du caractères 'c' par une simple occurrence
cut	-f x	extrait la colonne numéro x (le séparateur de colonnes par défaut est la tabulation)
sort	*(none)*	tri les lignes d'un fichier
uniq	*(none)*	supprime les lignes identiques

Table 2 – Opérateurs de redirection

Par exemple, la commande suivante permet d'afficher les 10 premières lignes retournées par la commande **ls -l /usr/bin** :

```
$ ls -l /usr/bin | head -n 10

-rwxr-xr-x 1 root   root          39 Jun 14 18:50 7z
-rwxr-xr-x 1 root   root          40 Jun 14 18:50 7za
-rwxr-xr-x 1 root   root      106520 May 16 13:48 a2p
lrwxrwxrwx 1 root   root          52 Sep 17 06:21 a2ping -> ../share/texliv
-rwxr-xr-x 1 root   root         883 Feb 23  2007 a5booklet
lrwxrwxrwx 1 root   root          54 Sep 17 06:21 a5toa4 -> ../share/texliv
lrwxrwxrwx 1 root   root          25 Sep 21 01:15 aclocal -> /etc/alternati
-rwxr-xr-x 1 root   root       36792 Aug 15 12:37 aclocal-1.15
-rwxr-xr-x 1 root   root       18872 Jul  2 14:31 aconnect
-rwxr-xr-x 1 root   root       19760 Nov  5  2013 acpi
```

La commande suivante permet de remplacer les espaces multiples dans la sortie de **ls** par un espace unique :

```
$ ls -l /usr/bin/ | head -n 10 | tr -s ' '

-rwxr-xr-x 1 root root 39 Jun 14 18:50 7z
-rwxr-xr-x 1 root root 40 Jun 14 18:50 7za
-rwxr-xr-x 1 root root 106520 May 16 13:48 a2p
lrwxrwxrwx 1 root root 52 Sep 17 06:21 a2ping -> ../share/texlive/texmf-dis
-rwxr-xr-x 1 root root 883 Feb 23 2007 a5booklet
lrwxrwxrwx 1 root root 54 Sep 17 06:21 a5toa4 -> ../share/texlive/texmf-dis
lrwxrwxrwx 1 root root 25 Sep 21 01:15 aclocal -> /etc/alternatives/aclocal
-rwxr-xr-x 1 root root 36792 Aug 15 12:37 aclocal-1.15
-rwxr-xr-x 1 root root 18872 Jul 2 14:31 aconnect
-rwxr-xr-x 1 root root 19760 Nov 5 2013 acpi
```

Exercices

1. Écrivez la commande permettant de remplacer les espaces de la sortie de **`ls`** par une tabulation '`\t`' ;
2. Écrivez la commande permettant d'extraire le nom des groupes propriétaires des fichiers présents dans `/usr/bin` (utilisez la commande de la question précédente) ;
3. Écrivez la commande permettant de trier le nom des groupes propriétaires des fichiers présents dans `/usr/bin` ;
4. Écrivez la commande permettant d'obtenir la liste des noms des groupes propriétaires des fichiers présents dans `/usr/bin` ;
5. Peut-on se passer de la commande de la question 1 ? Si oui, comment ?