Méthodes Empiriques et Langages de Script
TP 1

Gabriele A. Musillo
musillo4@etu.unige.ch

November 9, 2005

1 Exercice 1: le shell BASH

- Imaginez que vous ayez éxécuté les commandes `cd`, `ls -l`, `ps` et `emacs tsort.pl &` dans cet ordre. Quelle est la plus courte commande qui vous permet de relancer la commande `ls -l` ?
- Vous êtes dans votre *home*. Vous désirez créer les sous-répertoires `dir1/dir2`. Quelle est la commande et son option la plus appropriée qui crée ces répertoires ?
- Décrivez en termes de flots d'entrée, de sortie ou d'erreur les commandes suivantes:
  - `cmd > file`
  - `cmd 1> file`
  - `cmd 2> file`
  - `cmd > file 2>&1`
  - `cmd > file1 2> file2`
  - `cmd 2>> file`
  - `cmd >> file 2>&1`
  - `cmd1 2>&1 | cmd2`
  - `cmd1 | tee file_cmd1 | cmd2 | cmd3 > file1`
- Décrivez l'option `-p` de la commande `cp`.
- Décrivez l'option `-C` de la commande `ps`.
- Donnez deux noms de fichiers décrits par la wildcard `??[!1-5]`.
- Les expressions régulières `[A-Z]*` et `^[A-Z]*$` sont-elles équivalentes pour `grep -E` ?

- Vous désirez supprimer le répertoire /home/rmme et ce qu'il contient. Quelle commande et quelles options utiliseriez-vous ?
- Lesquelles des wildcards ci-dessous matchent les noms `Linux` et `linux`, mais ne matchent pas les noms `Llinux` et `Linux1` ?
  - `[L/linux]`, `?inux`, `\L\linux`, `[Ll]inux`, `[lL]inux?`
- Comment afficher la 27ème ligne d'un fichier ? Formulez deux solutions.
- Vous cherchez à terminer un processus dont le pid est 4077. Vous entrez la commande `kill 4077`. Mais le processus n'est pas encore mort. Quel signal doit-il lui être envoyé pour le tuer ?
- Vous avez lancé `emacs` et vous cherchez à tuer son processus. Mais vous ignorez son pid. Dans ces conditions, quelle est la commande qui le tuerait ?

2 Exercice 2: find

- Parcourez les pages `man` de la commande `find` et décrivez les commandes suivantes:
  - `find . -type f \( -name "*.c" -o -name "*.sh" \)`
  - `find bin/ -name "*.pl" |xargs chmod u+x`

3 Exercice 3: un script BASH

Considérez le script suivant et décrivez-le:

```
#!/bin/bash
#USAGE: regex_cut <pattern> <file> <delim>
cut -d "$3" -f $(head -n 1 $2 |tr "$3" '\n' |nl |egrep $1 |cut -f 1 |head -1) $2
```

Indications: les variables `$1`, `$2` et `$3` dénotent respectivement les arguments `<pattern>`, `<file>`, `<delim>`; l'instruction `$(cmd)` renvoie la sortie de la commande `cmd` (par example, `echo $(ls)` affiche la sortie de la commande `ls`).