INSTITUT Mines-Télécom

TELECOM SudParis

CI3 – Système de Fichiers : Outils incontournables

CSC3102 – Introduction aux systèmes d'exploitation

Elisabeth Brunet

28/09/2015

Plan

- Méta-caractères

- Outils incontournables
  - Nature d'une entrée
  - Pour les fichiers texte : affichage, tri, recherche de motif
  - Occupation disque
  - Archivage de fichiers
  - Recherche de fichiers

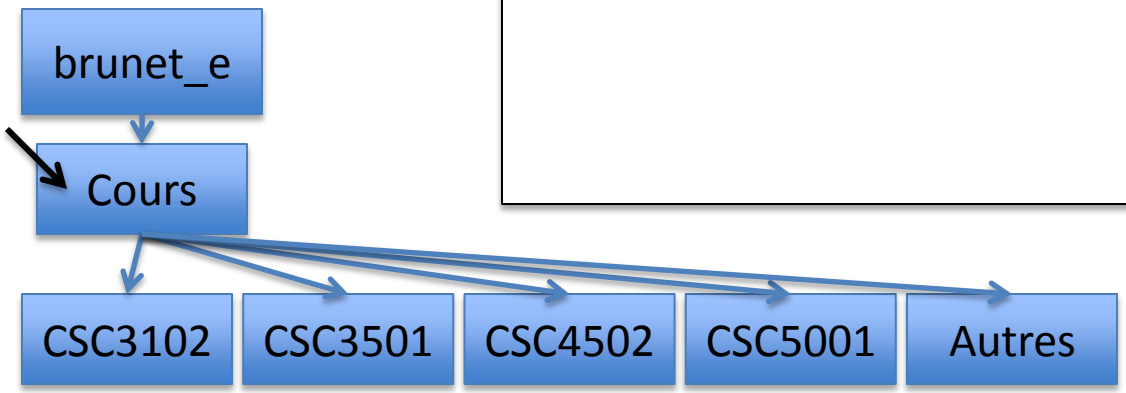
Méta-caractères dans les noms d'entrées du système de fichiers (1/2)

- Permet de générer un ensemble de chaînes de caractères
  - Pour généraliser un traitement sur un ensemble d'entrées
- Substitution par une valeur quelconque
  - `*` → une chaine de caractères quelconque (même vide)
  - `?` → substitue **un** caractère quelconque

Méta-caractères dans les noms d'entrées du système de fichiers (1/2)

- Permet de générer un ensemble de chaînes de caractères
  - Pour généraliser un traitement sur un ensemble d'entrées
- Substitution par une valeur quelconque
  - `*` → une chaine de caractères quelconque (même vide)
  - `?` → substitue **un** caractère quelconque
  - Exemple :

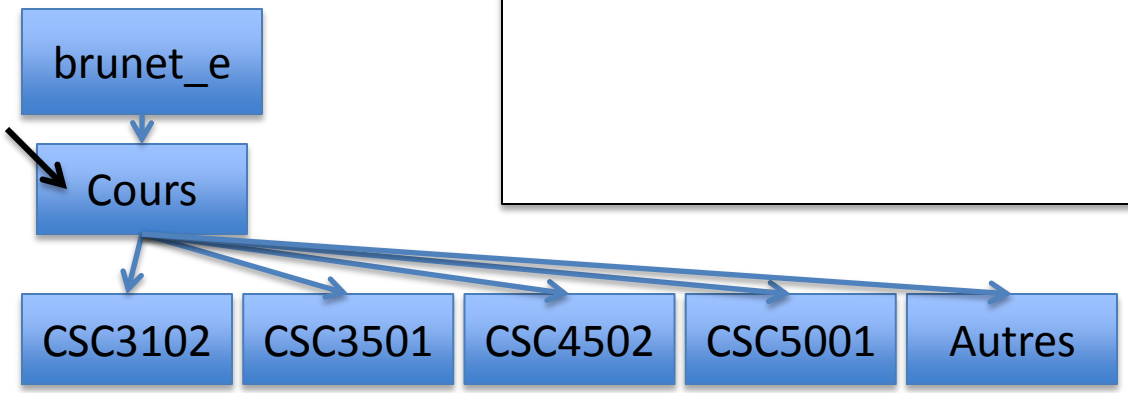
```
$cd ~/Cours
$
```



Méta-caractères dans les noms d'entrées du système de fichiers (1/2)

- Permet de générer un ensemble de chaînes de caractères
  - Pour généraliser un traitement sur un ensemble d'entrées
- Substitution par une valeur quelconque
  - `*` → une chaine de caractères quelconque (même vide)
  - `?` → substitue **un** caractère quelconque
  - Exemple :

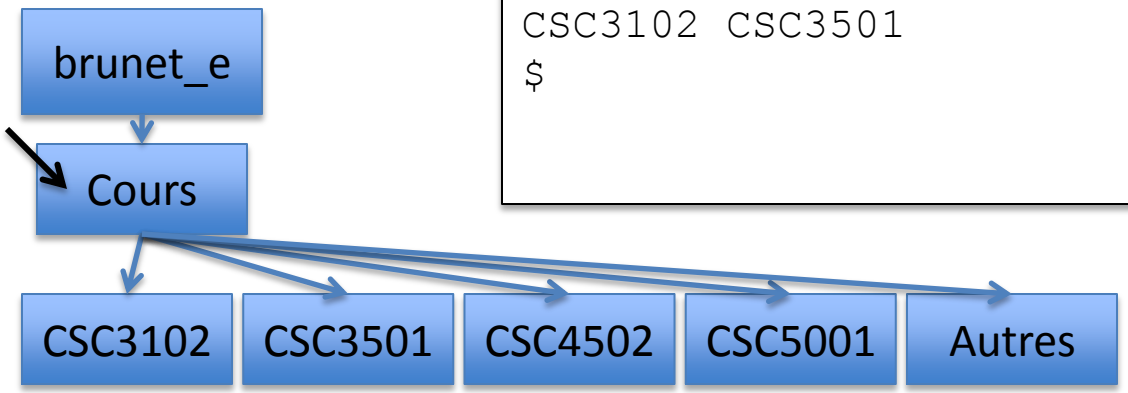
```
$cd ~/Cours
$ls -d CSC*           #tous les cours
CSC3102 CSC3501 CSC4502 CSC5001
$
```



Méta-caractères dans les noms d'entrées du système de fichiers (1/2)

- Permet de générer un ensemble de chaînes de caractères
  - Pour généraliser un traitement sur un ensemble d'entrées
- Substitution par une valeur quelconque
  - `*` → une chaine de caractères quelconque (même vide)
  - `?` → substitue **un** caractère quelconque
  - Exemple :

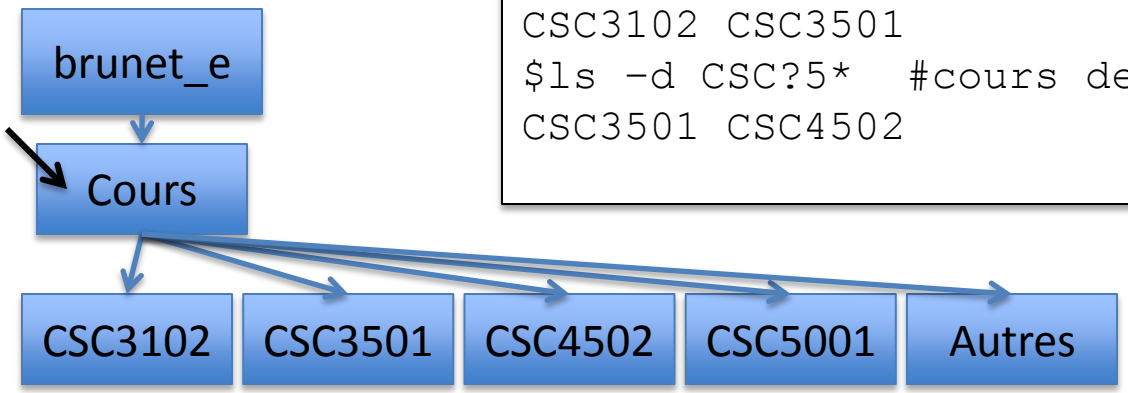
```
$cd ~/Cours
$ls -d CSC*           #tous les cours
CSC3102 CSC3501 CSC4502 CSC5001
$ls -d CSC3?0?  #cours de 1ère année
CSC3102 CSC3501
$
```



Méta-caractères dans les noms d'entrées du système de fichiers (1/2)

- Permet de générer un ensemble de chaînes de caractères
  - Pour généraliser un traitement sur un ensemble d'entrées
- Substitution par une valeur quelconque
  - `*` → une chaine de caractères quelconque (même vide)
  - `?` → substitue **un** caractère quelconque
  - Exemple :

```
$cd ~/Cours
$ls –d CSC*            #tous les cours
CSC3102 CSC3501 CSC4502 CSC5001
$ls –d CSC3?0?  #cours de 1ère année
CSC3102 CSC3501
$ls –d CSC?5*  #cours de 2nd semestre
CSC3501 CSC4502
```

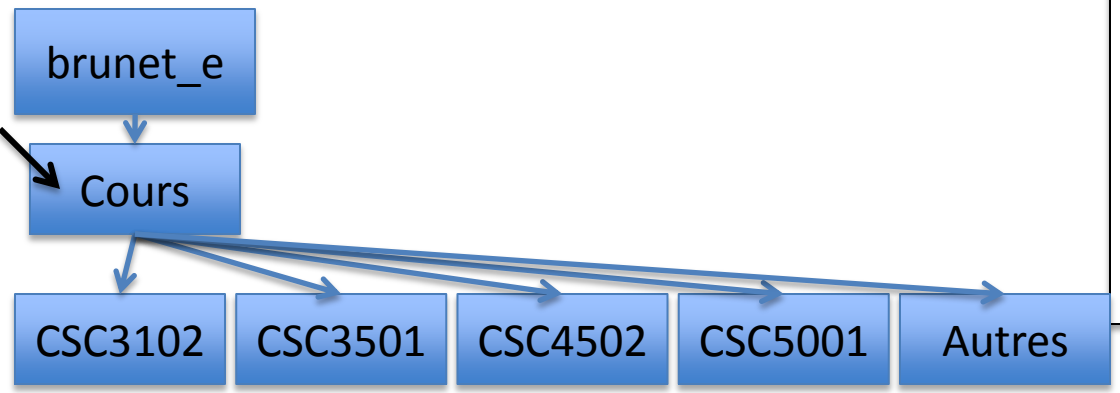


Méta-caractères dans les noms d'entrées du système de fichiers (2/2)

- Substitution prise dans un ensemble
  - `[...]` → substitue un caractère dans l'ensemble donné
  - `[!...]` → substitue un caractère hors de l'ensemble donné
- Ensemble
  - Liste de caractères : par ex., `[abc]`
  - Un intervalle : par ex., `[0-9]`
  - Ensembles prédéfinis :
    - `[[:alpha:]]` → caractères alphabétiques
    - `[[:lower:]]` / `[[:upper:]]` → alphabet minuscule / majuscule
    - `[[:digit:]]` → chiffres décimaux `[0-9]`

Méta-caractères dans les noms d'entrées du système de fichiers (2/2)

- Substitution prise dans un ensemble
  - `[...]` → substitue un caractère dans l'ensemble donné
  - `[!...]` → substitue un caractère hors de l'ensemble donné
- Ensemble
  - Liste de caractères : par ex., `[abc]`
  - Un intervalle : par ex., `[0-9]`
  - Ensembles prédéfinis :
    - `[[:alpha:]]` → caractères alphabétiques
    - `[[:lower:]]` / `[[:upper:]]` → alphabet minuscule / majuscule
    - `[[:digit:]]` → chiffres décimaux `[0-9]`
- Exemple

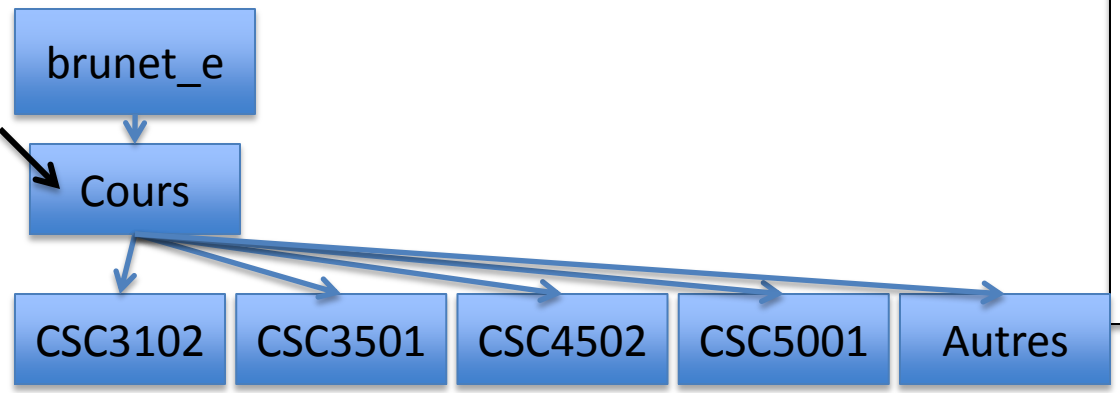



```
$cd ~/Cours
$
```

Méta-caractères dans les noms d'entrées du système de fichiers (2/2)

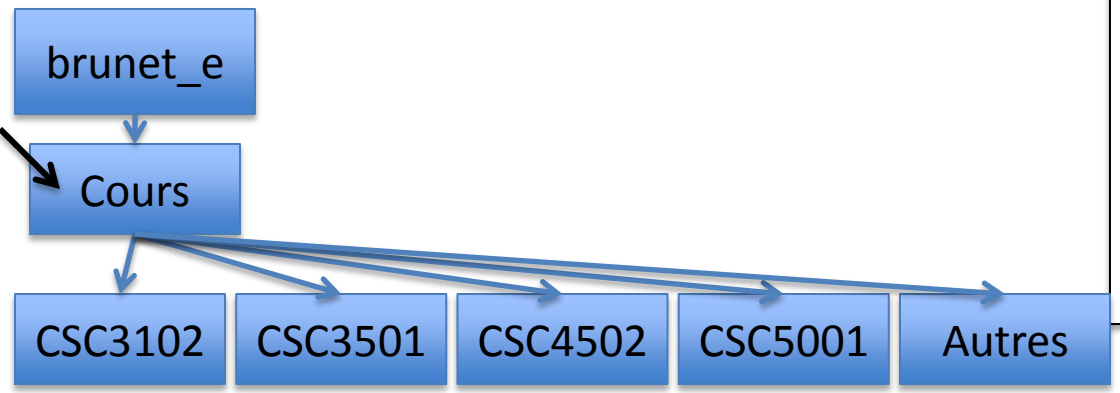
- Substitution prise dans un ensemble
  - `[...]` → substitue un caractère dans l'ensemble donné
  - `[!...]` → substitue un caractère hors de l'ensemble donné
- Ensemble
  - Liste de caractères : par ex., `[abc]`
  - Un intervalle : par ex., `[0-9]`
  - Ensembles prédéfinis :
    - `[[:alpha:]]` → caractères alphabétiques
    - `[[:lower:]]` / `[[:upper:]]` → alphabet minuscule / majuscule
    - `[[:digit:]]` → chiffres décimaux `[0-9]`
- Exemple

```
$cd ~/Cours
$ls -d CSC[45]* #cours de 2A et 3A
CSC4502 CSC5001
$
```



Méta-caractères dans les noms d'entrées du système de fichiers (2/2)

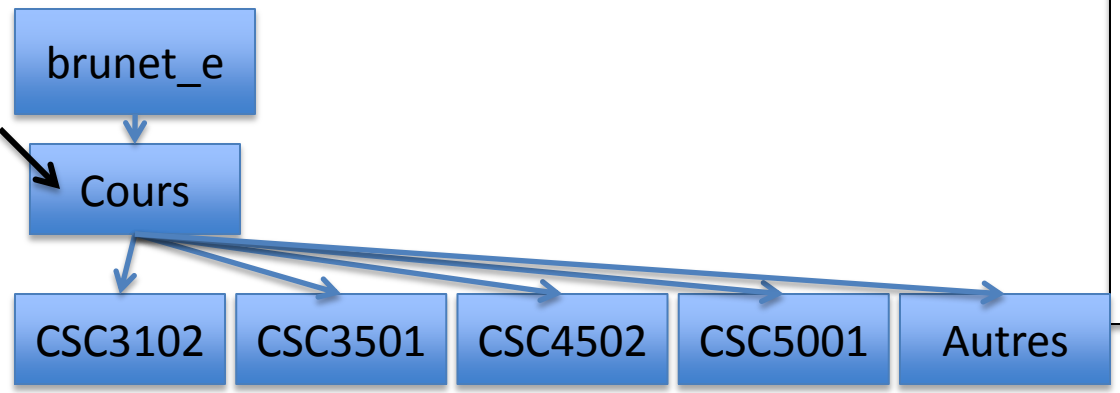
- Substitution prise dans un ensemble
  - `[...]` → substitue un caractère dans l'ensemble donné
  - `[!...]` → substitue un caractère hors de l'ensemble donné
- Ensemble
  - Liste de caractères : par ex., `[abc]`
  - Un intervalle : par ex., `[0-9]`
  - Ensembles prédéfinis :
    - `[[:alpha:]]` → caractères alphabétiques
    - `[[:lower:]]` / `[[:upper:]]` → alphabet minuscule / majuscule
    - `[[:digit:]]` → chiffres décimaux `[0-9]`
- Exemple




```
$cd ~/Cours
$ls -d CSC[45]* #cours de 2A et 3A
CSC4502 CSC5001
$ls -d CSC[!3]*  #cours pas de 1A
CSC4502 CSC5001
$
```

Méta-caractères dans les noms d'entrées du système de fichiers (2/2)

- Substitution prise dans un ensemble
  - `[...]` → substitue un caractère dans l'ensemble donné
  - `[!...]` → substitue un caractère hors de l'ensemble donné
- Ensemble
  - Liste de caractères : par ex., `[abc]`
  - Un intervalle : par ex., `[0-9]`
  - Ensembles prédéfinis :
    - `[[:alpha:]]` → caractères alphabétiques
    - `[[:lower:]]` / `[[:upper:]]` → alphabet minuscule / majuscule
    - `[[:digit:]]` → chiffres décimaux `[0-9]`
- Exemple




```
$cd ~/Cours
$ls -d CSC[45]* #cours de 2A et 3A
CSC4502 CSC5001
$ls -d CSC[!3]*  #cours pas de 1A
CSC4502 CSC5001
$ls *[[:alpha:]]
Autres
```

Nature d'une entrée du système de fichiers

- Traitement applicable à un fichier dépend de sa nature
  - Est-ce un fichier texte? Une image? Une archive? Un pdf?
- Commande `file` : affiche la nature d'une entrée
  - Si texte, précise le type d'encode
    - ASCII s'il n'y a que des caractères, UTF-8 si caractères accentués, etc.

```
$ file *
TP3              directory
TP3.html     exported SGML document, UTF-8 Unicode text
Ci3.pdf        PDF document, version 1.5
Ci3.pptx      Microsoft Powerpoint 2010
Notes.txt    ASCII text
Pedagogie.txt    UTF-8 Unicode text
```

Affichage d'un fichier en mode texte

- Consultation du contenu d'un fichier ordinaire

- `more <fichier>`
- `less <fichier>`

} affichage simple page par page

- `head [-n] <fichier>` : affichage des n premières lignes
- `tail [-n] <fichier>` : affichage des n dernières lignes

- `cat <fic1> [fic…]` : affiche la concaténation des fichiers indiqués

- `wc [-l] [-w][-c] <fic>` : compte les lignes, mots et caractères du fichier
  - Option `-l`, compte uniquement les lignes ; `-w`, les mots ; `-c`, les caractères

Trier les lignes de fichiers texte

- `sort [-n][-k][-t][-r] <fichier> [fic…]`
  - Par défaut, tri lexicographique
    - Option `-n` pour un tri numérique
  - Par défaut, tri appliqué en tenant compte de toute la ligne
    - Option `-k x[,y]` pour un tri selon les champs `x` à `y`
      - `sort -k 2 fichier` : tri selon le 2ème champ de chaque ligne
      - `sort -k 2,4 fichier` : tri selon les 2 à 4ème champs de chaque ligne
  - Par défaut, le séparateur de champs est l'espace
    - Option `-t <caractère>` pour changer le séparateur
  - Option `-r` pour inverser l'ordre du tri appliqué
  - Peut s'appliquer sur un ensemble de fichiers
  - D'autres options à consulter dans la page de manuel

Recherche d'un motif dans un fichier texte

- `grep [-v][-r] <motif> <fichier> [fic …]`
  - Affiche les lignes des fichiers contenant le motif
  - Le motif est une expression régulière (ou rationnelle)
    - `grep` = global regular expression print
    - ➢ Pour CSC3102, seul un sous-ensemble d'expressions régulières
    - Chaînes de caractères
    - Attention! Les méta-caractères vus précédemment n'ont pas la même signification dans les expressions régulières!
      - `.` : n'importe caractère
      - `?` / `*` / `+` : répétition du caractère précédent 0 ou une fois / 0 ou plusieurs fois / au moins une fois
      - `[…]` (/ `[^…]`) : met en correspondance un caractère de (/hors) l'ensemble
  - Option `-v` affiche les lignes ne contenant pas le motif
  - Option `-r` permet de chercher récursivement le motif dans la sous-arborescence du répertoire donné en argument
  - D'autres options à consulter dans la page de manuel

Taille de l'occupation disque

- `df` : connaître l'état d'occupation des partitions
- `ls -lh […]` : taille des entrées cible
  - Si répertoire, donne la taille nécessaire au stockage de sa table d'entrées mais n'inclut pas celle de ses sous-entrées
  - Si lien symbolique, donne sa taille, i.e. l'espace nécessaire au stockage dû chemin vers sa cible, ce qui correspond au nombre de caractères de ce chemin
- `du` : totalise l'occupation disque d'une entrée
  - Si répertoire, parcours récursif de son arborescence
  - Par défaut, donne le nombre de blocs occupés
    - Option `-h`, pour afficher l'équivalent de ce nombre de blocs de manière « *lisible pour un humain* » en o/K/M/G

Archivage

- Rassemblement d'une arborescence de fichiers en un seul fichier
- Commande `tar` (pour **t**ape **ar**chive)
  - Pour créer : `tar -c -v -z -f <nom_archive.tgz> <rep>`
    - Option `-c`, pour créer l'archive
    - Option `-v`, pour un affichage en mode verbeux
    - Option `-z`, pour une compression des données au format gzip
    - Option `-f`, pour préciser le nom de l'archive voulue
    - Par convention, le nom de l'archive porte l'extension `.tgz` ou `.tar.gz`
  - Pour extraire : `tar -x -v -z -f <nom_archive.tgz>`
    - Option `x`, pour extraire; `z`, pour la décompression via gzip
    - Décompression dans le répertoire courant

Recherche dans une arborescence

- `find` : recherche de entrées satisfaisant un ensemble de critères de sélection dans une arborescence
  - Parcourt récursivement et teste les critères à chaque étape
  - `find <rep_de_recherche> <liste des critères>`
    - `-name <chaine>` : précise le nom des entrées à rechercher
      - Peut inclure des méta-caractères
    - `-print` : permet l'affichage des résultats
    - ex: `find . -name core -print`

      → affiche les chemins des entrées nommées core de mon répertoire courant

      `find /usr -name ''*.c '' -print`

      → affiche les chemins des entrées dont le nom se terminent par `.c` sous `/usr`

- `locate` : recherche un fichier dans les répertoires d'installation du système d'exploitation
  - ex : `locate passwd`

Conclusion

- Concepts clés :
  - Méta-caractères : caractères jokers + ensembles

- Commandes clés :
  - `more`, `less`, `head`, `tail`, `cat`, `wc`
  - `sort`
  - `grep`
  - `df`, `du`, `ls -lh`
  - `grep`
  - `tar`
  - `find`, `locate`

En route pour le TP!