

Fichiers et systèmes de fichiers

Fichier

- Une suite plate de caractères ou une suite plate de *bits* (0 vs. 1). Un fichier a donc une taille
- Chaque application (sup)pose une certaine structure sur cette suite plate
- Ex. : un éditeur (même simple) utilise certains caractères pour découper en lignes

Systemes de fichiers 1/5

- Un arbre (ou un ensemble de boîtes enchâssées),
 - racine unique
 - un chemin et un seul d'un noeud ou d'une feuille à l'autre
- Embranchement d'un arbre ('répertoire' – *directory* – ou 'dossier') : on y trouve le catalogue des embranchements et fichiers qui sont dominés par cet embranchement
- Facettes d'un embranchement ou d'un fichier :
 - propriétaire (droits)
 - taille
 - date de création/modification
 - programme exécutable ou non
- Raccourcis dans l'arbre (grimper dans les arbres est fatiguant) : 'raccourci', alias, 'lien symbolique' (comme les échelles du jeu de l'oie)

Systemes de fichiers 2/5

- Localiser un fichier/un répertoire
 - de manière absolue, en décrivant les étapes du chemin (unique) par rapport à la racine
 - de manière relative : en décrivant le cheminement par rapport à l'endroit courant
- Chemin absolu : cygwin.exe, partir du disque C :, prendre l'embranchement Program files, puis l'embranchement cygwin
- Chemin relatif : pour aller du dossier où figure cygwin à celui où figure lexico, remonter dans le dossier-père et prendre l'embranchement Lexico3

Quand on déplace un dossier et ce qu'il englobe, les chemins entre dossiers/fichiers de ce dossier doivent être relatifs, sinon ils ne seront plus valables.

Systemes de fichier 3/5

- Se déplacer/ se situer par un simulacre graphique
- vs. dire textuellement qu'on veut se déplacer ou savoir où l'on est (interprète de commandes)

`pwd` (print working directory)

`ls` (list) et `ls -la` (pour avoir certaines propriétés des fichiers/embranchements)

```
cd /home/habert/enseignement/Plurital
```

```
cd ../ParisX/DEFI
```

Systemes de fichier 4/5

- Contenu d'un fichier
`cat <nom du fichier>`
- Copier un fichier
`cp <fichier1><fichier2>`
- Supprimer la référence à un fichier
`rm <nom du fichier>`
- Création d'embranchement et de son catalogue
`mkdir <nom de repertoire>` (et `rmdir <nom de repertoire vide>`)

Systemes de fichier 5/5

- Certaines parties de l'arborescence sont toujours présentes sur la machine (en particulier celles qui contiennent le système d'exploitation, le programme qui permet aux autres de fonctionner)
- D'autres parties de l'arborescence sont reliables temporairement (*montage*) : CDRom, clés USB, appareils photos USB, etc. Deux possibilités :
 - la « greffe » temporaire est transparente (auto-montage)
 - la « greffe » suppose une intervention plus ou moins importante

Transformer un fichier

2 approches

- « en bloc »
 - $\emptyset + \text{logiciel} = \text{fichier}$ (je lance le composeur Mozilla pour créer un document « à partir de rien »)
 - $\text{fichier}_1 + \text{logiciel} = \text{fichier}_1'$ (je lance Word pour modifier un fichier Word existant)
- « raffinement graduel » d'un flux de caractères : chaque étape hérite d'une version transformée de l'étape précédente
 - Sous Unix/Linux, le « passage de relais » s'effectue par la barre verticale (dit *pipe*)
 - Ex. : flux de caractères de départ | raffinement₁ | raffinement₂ | ... | raffinement_n

Ecrire (dans) un fichier

- Sous un logiciel

- En *redirigeant* un flux de caractères sous Unix/Linux

```
cat DescriptionTroisGorges.txt | grep  
'barrage' > DescriptionTroisGorges.barrage
```

```
cat LaFemmeEstUnBarrage.txt | grep  
'barrage' > LaFemmeEstUnBarrage.barrage
```

- On peut ajouter dans un fichier

```
cat DescriptionTroisGorges.txt | grep  
'barrage' > SensBarrage1
```

```
cat LaFemmeEstUnBarrage.txt | grep  
'barrage' >> SensBarrage1
```