Université de Nice-Sophia Antipolis
Polytech' CiP1

Jeudi 1er février 2007
Durée: 1h30

Création et Manipulation de documents

(Hélène Renard / Sylvain Schmitz)

Travaux Dirigés – Séance n^{o}1

---

1 Objectifs du TD

Formats de document. Vous allez voir au cours de ce TD qu'un document obéit à un ensemble de règles de format, qui sont respectées par toutes les applications amenées à le lire et/ou à l'écrire.

2 Notion de fichier

Un fichier est une unité informatique contenant des informations numériques. Cet ensemble structuré est stocké sur un support informatique.

Techniquement, c'est une séquence de bits formant une entité identifiée par une dénomination (le chemin dans un système de fichiers, une adresse Internet, *etc.*). Un fichier peut être le code exécutable d'une application ou d'une bibliothèque, un dossier ou un document. Ce qui donne un sens à un fichier est son *format*. La manipulation d'un fichier par des applications différentes nécessite un format commun.

2.1 Format de fichier

Le format d'un fichier correspond à la façon dont sont enregistrées les données. En effet, lorsque vous enregistrez un texte, une image ou tout autre document numérique, ce que vous enregistrez ne sera jamais qu'une suite de `0` et de `1`. La connaissance du format employé donne comment interpréter cette séquence de `0` et de `1`.

Quelques exemples de formats (voir `http://en.wikipedia.org/wiki/Category:Computer_file_formats`) :

texte brut désigne le format de fichier qui suit un encodage de caractères (*cf.* section 4.1), qui traduit le flux binaire de `0` et `1` en une séquence de caractères. Un fichier en texte brut peut être par ailleurs structuré pour respecter un *format texte* (source Java, document XML, *etc.*) ; par opposition, un format qui ne se décode pas en caractères est qualifié de *format binaire*.

ELF (*Executable and Linkable Format*) est le format binaire employé pour le code exécutable (objets, applications, bibliothèques) de la plupart des systèmes Unix actuels.

XML (*eXtensible Markup Language*) est un format texte de structuration de données crée pour faciliter les échanges entre applications, particulièrement au travers d'Internet.

PDF (*Portable Document Format*) est un format binaire pour les documents en deux dimensions.

OpenDocument un dialecte XML pour les documents bureautiques.

2.2 Formats standards

On peut enfin distinguer les formats selon le degré d'ouverture de leur spécification :

1. Formats gérés par un organisme normatif (ISO, ECMA, ANSI, IETF, W3C, *etc.*), avec une spécification rendue publique (HTML, OpenDocument). On parle dans ce cas de format *standard*. Attention, certains standards sont couverts par des brevets qui en limitent l'utilisation (MP3, GIF).
2. Formats propriétaires ouverts, dont la spécification est publique (PDF).
3. Formats propriétaires fermés, dont la spécification et l'utilisation est parfois monnayable (format des documents MS Word).

3 Reconnaître le format d'un fichier

Comment connaître le format d'un document ? Par son extension, par son type MIME, par son en-tête.

3.1 Extensions

Le format impose souvent un nom d'extension (`.pdf` pour PDF, `.html` pour HTML, *etc.*). L'extension est habituellement la partie du nom du fichier comprise entre le dernier point et la fin du fichier.

Les applications peuvent donc se baser sur ce nom d'extension pour déterminer le format d'un fichier. C'est un moyen rapide et facile de filtrer le contenu d'un répertoire pour ne garder que les fichiers d'un format précis.

Exercice n°1 : Comment n'afficher que les scripts shell du répertoire `/usr/bin` en se basant sur les extensions ? Combien y en a-t'il ?

Attention cependant, car ce n'est jamais qu'un nom, qui ne correspond pas toujours au format réellement employé. Un utilisateur qui fait aveuglément confiance à l'extension d'un fichier attaché à un mail risque de mauvaises surprises...

3.2 Types MIME

Les types MIME (*Multi-purpose Internet Mail Extensions*) constituent une façon normalisée de classer les différents types de fichiers sur Internet. Les programmes Internet, comme les serveurs et les navigateurs Web, comprennent tous une liste de types MIME afin de pouvoir transférer les mêmes types de fichiers de la même façon, quel que soit le système d'exploitation qu'ils utilisent.

Le type MIME est utilisé d'une part pour typer les documents attachés à un courrier mais aussi pour typer les documents transférés par le protocole HTTP. Ainsi lors d'une transaction entre un serveur Web et un navigateur internet, le serveur Web envoie en premier lieu le type MIME du fichier envoyé au navigateur, afin que ce dernier puisse savoir de quelle manière afficher le document. Ces types sont lisibles dans `/etc/mime.types`.

Si vous visualisez le fichier `/etc/mime.types`, vous allez voir qu'un type MIME comprend deux parties : un type et un sous-type. Ils sont séparés par une barre oblique (/). Par exemple, le type MIME des fichiers Word de Microsoft est `application` et leur sous-type est `msword`. Au complet, le type MIME est `application/msword`. De même, le type MIME d'une image GIF est `image/gif`.

Exercice n°2 : Quel est le type MIME d'un document texte brut ?

Comme pour les extensions de fichier, cette information n'est pas toujours fiable (un serveur pourrait mentir sur le type réel du fichier servi).

3.3 En-têtes de fichier

Le moyen le plus fiable est d'utiliser les en-têtes des fichiers. En effet, la plupart des formats mettent en début de fichier les informations sur le format employé.

Exercice n°3 : Comment regarder les premières lignes d'un fichier sous Unix ? Regardez l'en-tête de différents fichiers.

Il existe une commande Unix qui analyse l'en-tête des fichiers et la compare aux formats courants pour déterminer exactement à quel format est le fichier.

Exercice n°4 :

1. Commencez par regarder le manuel de la commande `file` et les différentes options qu'elle propose.
2. Placez vous dans votre Desktop, puis effectuez les commandes suivantes :
   - `file Trash`
   - `file *`
   - `file .*`

L'option `-z` de la commande `file`, lorsqu'elle est utilisé sur un fichier compressé, essaie d'analyser le type du fichier décompressé.

Par exemple :

```
$> file -z hasard.tar.gz
hasard.tar.gz: GNU tar archive (gzip compressed data, deflated,
last modified: Sun Sep 16 13:34:51 2006, os: UNIX)
```

Exercice n°5 : Cherchez tous les fichiers de type Bourne shell contenus dans le répertoire `/usr/bin` en vous fiant maintenant au résultat de la commande `file`. Combien y en a-t'il ?

4 Formats courants

4.1 Documents texte

Un codage de caractères est un code qui associe un jeu de caractères d'une langue naturelle avec un jeu de quelque chose d'autres, comme par exemple des nombres ou des signaux électriques. Le morse a été le premier codage à permettre une communication longue distance. Ce code est composé de points et de tirets (un codage binaire en quelque sorte...). Il permit d'effectuer des communications beaucoup plus rapides que ne le permettait le système de courrier de l'époque aux États-Unis. L'interpréteur était l'homme à l'époque, il fallait donc une bonne connaissance du code...

Dans les années 60, le code **ASCII** (*American Standard Code for Information Interchange*) est adopté comme standard. Il permet le codage de caractères sur 8 bits, soit 256 caractères possibles. Il est indispensable, pour l'échange d'information sur l'Internet, par exemple, de préciser le codage utilisé. Ne pas le faire peut transformer un document en un fouillis incompréhensible. Qui n'a jamais vu une page Web avec des points d'interrogation à la place des accents, ou un courriel avec des `=E9` au lieu des `é` ?

Le besoin de supporter de multiples écritures demande un nombre nettement plus élevé de caractères supportés que ce que nous avons vu jusqu'à maintenant et nécessite une approche systématique du codage de caractère utilisé. Le codage Unicode a pour ambition d'être un surensemble de tous les autres existants, et est souvent représenté en UTF-8.

4.1.1 UTF-8 : qu'est-ce que c'est ?

Unicode est une norme informatique, développée par le Consortium Unicode, qui vise à donner à tout caractère de n'importe quel système d'écriture de langue un nom et un identifiant numérique, et ce de manière unifiée, quelle que soit la plate-forme informatique ou le logiciel.

L'UTF-8 est le plus commun pour les applications Unix et Internet. Son codage de taille variable lui permet d'être en moyenne moins coûteux en occupation mémoire. Mais cela ralentit nettement les opérations où interviennent des extractions de sous-chaînes, car il faut compter les caractères depuis le début de la chaîne pour savoir où se trouve le premier caractère à extraire.

L'UTF-8 assure aussi, et c'est son principal avantage, une compatibilité avec les manipulations simples de chaînes en ASCII dans les langages de programmation. Ainsi, les programmes écrits en C peuvent souvent fonctionner sans modification.

4.1.2 Conversions de codage de caractères

- En ligne de commande : `iconv`.
- Sous Emacs : `M-x set-buffer-file-coding-system`.

Exemple :

```
iconv -f latin1 -t utf8 test.iso -o test.unic
```

convertit le fichier `test.iso` en `test.unic` de l'encodage ISO-LATIN-1 (aussi appelé ISO-8859-1, utilisé pour les langues latines !) en UTF-8.

4.1.3 Exercices

Exercice n°6 : Dans un shell, tapez la commande `man ascii` et regardez la table ASCII ainsi que les pages de manuel. Quel est le code ASCII (décimal) correspondant au caractère `A` ? Quel est le caractère correspondant au nombre 64 (décimal) ?

Exercice n°7 : Faites à présent `man iso-8859-1`. Quel est le caractère correspondant au nombre 233 ?

Exercice n°8 :

1. Créez un fichier contenant un court texte avec des accents. Si vous êtes en manque d'inspiration, cherchez ce poème sur le Web : *Le dormeur du val* et faites un copiez-le sous Emacs. Enregistrez ce fichier sous le nom `dormeur.txt`. Comment connaître le codage actuellement utilisé par Emacs ?
2. Changez le codage de caractères vers `iso-8859-1` sous Emacs, et enregistrez le fichier sous le nom `dormeur2.txt`. Vérifiez le codage employé avec `file`.
3. Changez `dormeur.txt` vers `latin1` avec `iconv` dans un fichier `dormeur3.txt`. Vérifiez avec `file`.

4.2 Archives compressées

4.2.1 Archivage

En informatique, une *archive* est un ensemble de fichiers réunis dans un seul fichier (l'archive), pour pouvoir plus facilement les sauvegarder ou les transmettre. Le fichier archive contient généralement une « table des matières » permettant de retrouver rapidement les fichiers contenus dans l'archive.

Les « archiveurs » sont des programmes permettant de créer et de manipuler les archives. On peut citer parmi les plus fréquemment utilisés les formats TAR, ZIP, RAR, ...

- Création d'une archive

```
tar cvf archive fichiers-ou-repertoires
c (créer) : pour fabriquer l'archive
v (verbeux) : utile pour suivre la progression de l'archivage
f (fichier) : archive créée dans un fichier (et non sur une bande)
```

Exemple : `tar cvf /backup/home.tar /home && bzip2 /backup/home.tar`

- Manipulation d'une archive

  - Afficher le contenu d'une archive ou vérifier son intégrité : `tar tvf archive`
  - Extraire tous les fichiers d'une archive : `tar xvf archive`
  - Extraire seulement quelques fichiers d'une archive : `tar xvf archive fichiers`, où les fichiers sont donnés avec un chemin relatif au répertoire racine de l'archive.

4.2.2 Compression

Les techniques de compression de données visent à réduire le volume des données pour économiser de l'espace mémoire (disque) ou accélérer les transmissions réseau. Ces techniques sont basées sur l'exploitation des redondances : en général, le contenu d'un fichier n'est pas complètement aléatoire.

1. Sous UNIX :
   - `compress`, `uncompress`, `zcat` : commandes standard UNIX
   - `gzip`, `gunzip` : GNU ZIP, versions libres s'utilisant exactement comme `compress` et `uncompress` mais plus performantes (rapidité et taux de compression)
   - `bzip2`, `bunzip2`
2. Sous DOS/WINDOWS : plusieurs utilitaires, gratuits ou commerciaux, le plus utilisé étant ZIP.

4.2.3 Exercices

Exercice n°9 :

1. Ouvrez votre navigateur et allez à l'URL `http://personal.riverusers.com/~thegrendel/`
2. Téléchargez l'archive `abs-guide-4.2.tar.bz2` (cliquez sur *Advanced Bash Scripting Guide* puis sur *The latest update of this document*).
3. Créez le répertoire `CMDocs/abs` et déplacez l'archive dans ce répertoire. Quelle est la taille de cette archive ?
4. Visualisez le contenu de l'archive sans la décompresser ni l'extraire.
5. Décompressez et extrayez l'archive en une seule commande.
6. Créez un répertoire de nom votre login.
7. Copiez sous ce répertoire tous les fichiers et sous-répertoires de `CMDocs/abs`.
8. Fabriquez une archive compressée avec `gzip` de ce répertoire (et non pas seulement du contenu de ce répertoire). Comment le faire avec une seule commande `tar` ? Quelle est la taille de cette archive ?

ATTENTION : ne décompressez jamais une archive, en particulier dans votre homedir, sans avoir auparavant listé son contenu, et vérifié que son contenu était bien dans un répertoire.

4.3 Formats d'images

4.3.1 Images vectorielles

Une image vectorielle est une représentation composée d'objets géométriques (lignes, points, polygones, courbes, ...) ayant des attributs de forme, de position, de couleur, *etc.*

Le principe de base d'une image vectorielle consiste à décrire des formes géométriques simples (arcs de cercle ou d'ellipse, segments de droite, courbes de Bézier...), auxquelles on peut appliquer différentes transformations : rotations, écrasement, mise à l'échelle. Les effets spéciaux permettent une grande souplesse : extrusion, effet miroir, dégradé de formes, morphage, etc.

Il existe de nombreux formats de fichiers vectoriels. On peut citer PostScript, EPS, PDF ou SVG.

4.3.2 Images scalaires

Il s'agit d'images pixellisées, c'est-à-dire un ensemble de points (pixels) contenus dans un tableau, chacun de ces points possédant une ou plusieurs valeurs décrivant sa couleur.

La qualité d'une image scalaire est déterminée par le nombre total de pixels (appelé sa définition) et la quantité d'information contenue dans chaque pixel.

Nous pouvons distinguer deux catégories d'images scalaires :

- compressées sans pertes, telles que TIFF et PNG. Le format PNG (*Portable Network Graphics*) vise à remplacer le format GIF qui n'est pas libre de droit. Une image en PNG sera 10% à 30% plus légère que celles en GIF.
- compressées avec pertes, telles que JPEG.

4.3.3 La commande `convert`

La commande `convert` permet de convertir le format d'une image en un autre format d'image mais aussi de redimensionner, recadrer, fusionner etc.

Quelques exemples d'utilisation :

- `convert rose.jpg rose.png`
- `convert rose.jpg -resize 50% rose.png`

Cette commande admet un grand nombre d'options, que vous trouverez en détail à l'URL `http://www.imagemagick.org/script/convert.php`

Exercice n°10 :

1. Créez le répertoire `CMDocs/img`.
2. Téléchargez une image PostScript (par exemple depuis `http://www.let.rug.nl/~kleiweg/postscript/`).
3. Déplacez-la dans le répertoire `CMDocs/img`.
4. Visualisez-la avec `evince` ; zoomez.
5. Changez son format en PNG et JPEG à l'aide de `convert`.
6. Comparez les résultats à l'image originale ; essayez de zoomer. Que constatez-vous ?