

## Cinquième partie V

### Exercice de découpage de problème et d'automatisation

#### Un album photo Web à partir du contenu d'une carte photo

### Plan

#### Énoncé du problème

Analyse du problème et découpage en tâches

Comment ranger, ordonner les photos et les catégories ?

Comment réaliser les opérations techniques, où trouver les informations ?

Let's go !

### Énoncé du problème

Un de vos amis décharge sur son ordinateur les photos prises au cours d'un séjour.

*Il a tout d'abord pris des photos de la gare à son arrivée, a vu une scène insolite en se rendant à son hôtel.*

*Le lendemain matin il est parti faire la visite du château recommandée par son guide et visité le jardin d'à côté. Il s'est ensuite rendu dans sa famille et a été très heureux de prendre moult photos de ses petits cousins et de l'arrière-grand-mère qu'il n'avait pas vu depuis fort longtemps.*

*En visitant le château, il a appris qu'une visite guidée spéciale des hauteurs du château et de parties usuellement non accessibles au public était organisée le lendemain matin, il s'y rend donc le lendemain. Ensuite il va déjeuner dans un village voisin et se balader sur la plage, puis revient voir le jardin botanique visité la veille et qui est éclairé le soir.*

## Énoncé du problème

Le lendemain matin il repart avec une carte photo bien pleine et il se promet de mettre ses photos sur le Web à disposition de quelques amis et de sa famille.

Aidez-le à fabriquer son album photo en regroupant les photos par catégories (la gare, la scène insolite, le château, le jardin botanique, les retrouvailles familiales, le village voisin et sa plage) sur une page d'index avec vignettes qui lorsqu'on clique dessus ouvre une page dédiée à cette photo avec un lien visuel sur la photo précédente et la photo suivante et un lien vers l'index de l'album.

## Énoncé du problème

photo-xxx .html

**Le titre de la catégorie**

La grande photo avec une bonne résolution

Lien vers l'index général

Les vignettes des photos précédentes avec un lien vers les pages de ces deux photos

## Énoncé du problème

index.html

**Photos prises de mon séjour à Petaouchnok au mois de novembre 2010**

**L'arrivée à la gare**

**Une scène insolite**

**Le château de Petaouchnok**

## Plan

Énoncé du problème

Analyse du problème et découpage en tâches

Comment ranger, ordonner les photos et les catégories ?

Comment réaliser les opérations techniques, où trouver les informations ?

Let's go !

## Analyse : traitement des photos

1. Répartition par catégorie et ordonnancement des catégories (travail humain)
2. Ordonnancement des photos dans chaque catégorie (travail humain)
3. Fabrication des vignettes (automatique)
4. Réduction de la taille des grandes photos (automatique)
5. Orientation des photos (automatique)

*Une fois la place de chaque photo déterminée (étapes 1 et 2), un script effectue sur chaque photo les étapes 3 à 5.*

## Analyse : génération des pages HTML

Fichiers à produire

- ▶ `index.html`
- ▶ `photo.html` pour chaque `photo.jpg`

*Un script qui parcourt les catégories puis les photos de chaque catégorie et produit pour chaque photo la page `photo.html` et la ligne de `index.html` correspondante.*

On aura certainement intérêt à écrire un script pour déterminer la photo précédente et la photo suivante pour simplifier la fabrication de la page Web dédiée à chaque photo.

# Plan

Énoncé du problème

Analyse du problème et découpage en tâches

**Comment ranger, ordonner les photos et les catégories ?**

Comment réaliser les opérations techniques, où trouver les informations ?

Let's go !

## Représentation des données

### Les catégories

- Commentaire Exif : ajoutable en ligne de commande, accessible textuellement, mais peu pratique
- Propriétés du fichier : difficile à utiliser, pas de moyen textuel d'y accéder, pas utilisable
- Utiliser l'arborescence Unix : le répertoire du fichier est la catégorie de la photo : visuellement ajoutable sous un gestionnaire de fichiers, textuellement ajoutable par des commandes de base Unix, textuellement visualisable par des commandes de base Unix, de façon rapide et évidente

Utiliser l'arborescence Unix pour représenter les catégories s'avère donc le choix évident

## Représentation des données

### L'ordre des catégories et des photos

- Soit l'ordre d'affichage des catégories et des photos par catégorie, est noté sur un bout de papier en attendant d'être décrit dans les scripts : les scripts sont difficiles à écrire, à lire et à réutiliser pour d'autres albums puisqu'ils sont parasités par cette information
- Soit l'ordre est fixé au niveau du système de fichiers et on peut avoir un traitement uniforme dans les scripts et réutiliser ces scripts pour la fabrication d'autres albums

C'est évidemment la seconde solution qu'on choisit.

## Représentation des données

### L'ordre des catégories et des photos

#### L'ordre est fixé au niveau du système de fichiers ?

- ▶ En parcourant les fichiers on lit aussi l'ordre dans lequel on doit les afficher.
- ▶ On va préfixer chaque catégorie (et à l'intérieur de chaque catégorie chaque photo) de son numéro d'affichage
- ▶ Par exemple
  - 01-Famille
  - 02-Chateau
  - 03-Jardin
  - 04-Petaouchnok
  - 05-Gare
  - 06-Scene insolite

C'est un peu fastidieux mais après on est tranquille.

## L'ordre des photos dans chaque catégorie

### L'ordre chronologique

- ▶ Les photos sont nommées par ordre chronologique par l'appareil photo et l'ordre chronologique correspond à l'ordre alphabétique : il suffit donc de parcourir le répertoire de la catégorie (par ordre alphabétique) pour parcourir les photos dans l'ordre où elles ont été prises.
- ▶ Par exemple les photos de la gare, de la scène insolite et du jardin seront affichées par ordre chronologique par un parcours simple.

## L'ordre des photos dans chaque catégorie

### Perturber l'ordre chronologique

- ▶ Pour certaines catégories en revanche on voudra modifier l'ordre d'affichage :
  - ▶ regrouper les photos de l'extérieur du château, puis de l'intérieur puis celles de la visite spéciale alors que le photographe a pris des photos de l'extérieur du château avant et après la première visite, voire en se baladant dans la ville.
  - ▶ lors de la visite du village voisin et de sa plage, il a pris des photos du village, puis est descendu à la plage où il a vu aussi des oiseaux, puis est remonté au village, il décide de séparer les photos du village, celles de la plage, puis celles des oiseaux.
  - ▶ lors de la visite familiale, l'aïeule est apparue dans l'après-midi après sa sieste mais il convient de placer les photos qui la concernent au début de l'album.

## L'ordre des photos dans chaque catégorie

### Perturber l'ordre chronologique

- Il est inutile de créer autant de numéros que de photos : on attribue le numéro 01 par exemple à une photo qu'on veut absolument placer en tête de sa catégorie, puis 02 à tout un groupe de photos qu'on veut voir immédiatement après et pour lequel l'ordre chronologique convient, puis 03 à une autre photo ou groupe de photos qu'on veut voir après le groupe 02, etc.
- L'affectation d'un numéro à tout un groupe de photos dans une catégorie s'automatise par une simple boucle `for` autour d'un `mv` mais la décision de la répartition en catégorie et l'ordonnancement relève de l'avis du photographe et ne peut être mécanisé.

## Plan

Énoncé du problème

Analyse du problème et découpage en tâches

Comment ranger, ordonner les photos et les catégories ?

Comment réaliser les opérations techniques, où trouver les informations ?

Let's go !

## Comment trouver l'info ?

### Première solution : `man -k`

```
% man -k photo
```

```
Image::ExifTool::PhotoMechanic (3pm) - Read/write Photo Mechanic informat
Image::ExifTool::Photoshop (3pm) - Read/write Photoshop IRB meta informat
Image::ExifTool::WritePhotoshop (3pm) - Write Photoshop IRB meta informat
c44 (1) - DjVuPhoto encode.
dcccancrw (1) - Canon (CRW) RAW photo file recovery utility
dcraw (1) - command-line decoder for raw digital photos
hp-unload (1) - Photo Card Access Utility
photorec (8) - Recover lost files from harddisk, digital camera a
sane-gphoto2 (5) - SANE backend for gphoto2 supported cameras
```

Rien de convaincant : interactivité inadaptée pour des scripts, travail sur des formats différents, etc.

```
% man -k image | wc -l
```

```
427
```

On est noyé sous l'information, impossible de trier ces 427 entrées, il faut des éléments pour filtrer l'information

## Comment trouver l'info ?

### Autres solutions

- ▶ le bouche à oreilles : « pour faire telle opération sur des photos, perso j'utilise tel outil... »
- ▶ le Web : sous google on cherche `photo Linux` et en regardant les premières pages on a un aperçu de ce qui existe et on voit que l'ensemble `ImageMagick` et particulièrement l'outil `convert` est l'outil à tout faire en ce qui concerne les manipulations d'images sur la ligne de commande
- ▶ ensuite il reste à comprendre comment utiliser `convert`...  
Oops ! il y a plus de 200 options... Vite un tutoriel sur le Web !  
Par exemple en quelques lignes dans le tutoriel Ubuntu d'`ImageMagick` on trouve ce qu'on cherche

## Réduction de la taille des grandes photos

`convert -resize geom image image_modifiee` où `geom` est de la forme `largeurxhauteur` en utilisant la commande en ligne `convert` du package `ImageMagick`.

Par exemple

```
% ls -l P1350207.JPG
-rwx----- 1 vmm vmm 3685170 2009-11-25 17:42 P1350207.JPG*
% convert -resize 1024x768 P1350207.JPG P1350207_resize.jpg
% ls -l P1350207_resize.jpg
-rw-rw-r-- 1 vmm vmm 97428 2009-11-25 22:27 P1350207_resize.jpg
```

## Vignettes

On aura recours au même outil pour la fabrication des vignettes

```
% convert -resize 4% P1350207.JPG P1350207.vignette.jpg
% ls -l P1350207.vignette.jpg
-rw-rw-r-- 1 vmm vmm 2982 2009-11-25 22:50 P1350207.vignette.jpg
```

## Orientation



```
<html>
<head>
<title> Orientation d'une photo de bouc </title>
</head>
<body>


</body>
</html>
```

## Orientation

On découvre dans le tutoriel Ubuntu que `convert` a une option `-auto-orient` :

- pour tourner les images en fonction de leur orientation donnée dans le format Exif, comme par exemple les images issues d'un appareil photo numérique. L'image n'est retournée que si elle a été prise en portrait, si elle a été prise en paysage, celle-ci n'est pas modifiée :

```
convert image.jpg -auto-orient image_retournee.jpg
```

C'est ce qu'on a utilisé pour fabriquer la seconde photo affichée

```
convert -auto-orient bouc_original.jpg bouc_debout.jpg
```

## La production du code HTML

### Sources d'information

- En librairie l'un des innombrables ouvrages qui traite de HTML.
- Des tutoriels pour HTML sur le Web. Citons <http://www.w3schools.com> le plus connu et le plus complet, où l'on peut faire des essais et en voir immédiatement le rendu.



## La production du code HTML

### Ce qu'il faut connaître

- Quelle est la structure d'une page html : les balises `<html>`, `<head>`, `<body>` et les fermantes correspondantes `</html>`, `</head>`, `</body>`
- à mi-chemin entre structure et apparence : les entêtes (balises `<h1>` à `<h6>`)
- créer un lien avec la balise `<a>` et son attribut `href`
- inclure une image avec la balise `<img>` et son attribut `src`

Pour faire des essais localement, on peut demander à voir n'importe quelle page HTML par

`file:///le/chemin/absolu/pour/trouver/le/fichier/html`  
dans son browser ou directement `firefox le/fichier/html`  
dans son répertoire.

## La production du code HTML

### Un exemple plus conséquent

```
<html>
<head>
<title> Une photo de bouc </title>
</head>
<body>
<h1 align="center">Un bouc dans une guérite</h1>

<h3>Liens textuels</h3>
<a href="prec.html"> Photo précédente</a>
<a href="succ.html"> Photo suivante</a>
<h3>Liens sur une image</h3>
<a href="prec.html"> </a>
<a href="succ.html"> </a>
</p>
</body>
</html>
```

## La production du code HTML

### ... et son résultat



## Rappel sur les documents en ligne

Il existe une construction en shell, appelé document en ligne, qui permet d'écrire dans un fichier du texte au kilomètre avec seulement l'expansion des variables par le shell :

|   |   |
|---|---|
| <pre>cat &gt; fichier &lt;&lt;EOF texte EOF</pre> | <pre>cat &gt;&gt; fichier &lt;&lt;EOF texte EOF</pre> |
|---|---|

Cette construction est particulièrement adaptée à la production de code HTML puisque celui-ci est très verbeux et comporte de nombreux caractères qui seraient interprétés par le shell lors d'une simple commande `echo` ou devraient être déspecialisés pour l'éviter, ce qui rendrait le code illisible.

## La production du code HTML

### Où placer la page ?

- Les fournisseurs d'accès donnent non seulement accès à Internet mais aussi à un serveur Web et un espace de stockage.
- On va sur le site de paramétrage de son compte et on trouve le nom de la page web réservée pour le photographe et les informations sur comment y placer des fichiers qui seront ensuite accessibles sur la page Web associée à son compte.

## La production du code HTML

### Comment rendre la page en question accessible à quelques personnes seulement ?

Si on place dans le même répertoire un fichier de nom `.htaccess`, on peut contrôler par ce fichier l'accès à cette page Web.

### Que faudra-t-il copier sur le site ?

Les fichiers au format HTML (celui d'index, point d'entrée de la page Web et ceux qui auront été macro-générés pour chaque photo), le `.htaccess`, les grandes photos réduites à la taille voulue ainsi que leurs vignettes.

## Plan

Énoncé du problème

Analyse du problème et découpage en tâches

Comment ranger, ordonner les photos et les catégories ?

Comment réaliser les opérations techniques, où trouver les informations ?

Let's go !

## Photos précédente et suivante

- En fait dans notre parcours des photos le nom de la photo précédente est gratuit : il suffit de le mémoriser avant de passer à la photo suivante. En revanche il faut travailler un peu plus pour connaître le nom de la photo suivante.
- Il nous faut récupérer le nom du fichier placé après dans la liste des fichiers de la catégorie et si on est au bout le premier fichier de la catégorie suivante.

## Photo suivante

Tout d'abord pour récupérer les noms du fichier suivant `$photo` du répertoire `$categorie`, on va énumérer les fichiers ligne par ligne avec l'option `-1` de la commande `ls` et extraire la ligne suivante celle du fichier `$photo` avec l'option `-A1` de la commande `grep` et il nous reste à extraire la dernière ligne avec `tail` :

```
SUIVANTE=`ls -1 $categorie | grep -A1 -F "$photo" | tail -n 1`
```

## Photo suivante

Alors, de deux choses l'une :

- ▶ soit on n'est pas à la fin de la catégorie et `$SUIVANTE` est le nom de la photo suivante,
- ▶ soit on est à la fin de la catégorie et alors la commande `grep` n'a pu rendre qu'une seule ligne, que `tail` a conservé et `$SUIVANTE` est égal à `$photo`, il nous faut alors aller chercher le nom de la catégorie suivante avec la même technique :

```
CAT_SUIVANTE=`ls -l | grep -A1 -F "$categorie" | tail -n 1`
```

puis le nom du premier fichier de la catégorie avec

```
SUIVANTE=`ls -l $CAT_SUIVANTE | head -n 1`
```

## Photo suivante

Il faut cependant faire attention car si l'on est à la dernière catégorie, alors `$CAT_SUIVANTE` est vide ou ne correspond pas à un répertoire de photos et il n'y a pas de photo suivante. On va donc affiner en

```
if echo $CAT_SUIVANTE | grep -q -E '^[0-9]'\n  then SUIVANTE=`ls -l $CAT_SUIVANTE | head -n 1`\n  else SUIVANTE=""\nfi
```

## Fonction photo\_suivante

Ce qui nous donne la fonction :

```
photo_suivante () {\n  local CAT_SUIVANTE\n  SUIVANTE=`ls -l $categorie | grep -A1 -F "$photo" | tail -n 1`\n  if [[ $photo == $SUIVANTE ]]\n  then CAT_SUIVANTE=`ls -l | grep -A1 -F "$categorie" | tail -n 1`\n    if echo $CAT_SUIVANTE | grep -q -E '^[0-9]'\n    then SUIVANTE=`ls -l $CAT_SUIVANTE | head -n 1`\n    else SUIVANTE=""\n    fi\n  fi\n}
```

- ▶ On choisit d'utiliser une fonction pour l'intégrer au script principal et partager l'environnement (`$categorie`, `$photo`, `$SUIVANTE`).
- ▶ En tout cas, on est content d'avoir isolé ce bout de code d'intendance, cela augmentera la lisibilité du code dont le but essentiel sera la production de code HTML.

## Structure du script principal

### Structure générale : une double boucle

- Pour parcourir toutes les photos, nous utiliserons une double boucle : une boucle sur les catégories puis à l'intérieur une boucle sur les photos de la catégorie.
- Avant ces boucles : l'écriture de l'entête du fichier `index.html`, la création des répertoires dans lesquels on stockera la photo réduite, la vignette et la page HTML de chaque photo créée dans la boucle.

```
cat > index.html <<EOF
<html>
<head><title>Photos prises de mon séjour à Petaouchnok au mois de
novembre 2010</title></head>
<body>
<h1 align="center"> Photos prises de mon séjour à Petaouchnok
au mois de novembre 2010</h1>
EOF
mkdir pages_photo photos vignettes
```

- Après les boucles : l'écriture du pied de page du fichier `index.html`.

```
cat >> index.html <<EOF
</body>
</html>
EOF
```

## Structure du script principal

### Les boucles et leur intendance

```
for categorie in [0-9]*; do # ne traiter que les répertoires de photos
```

```
# le vrai nom de la catégorie, sans le numéro, pour le titre
nom_categorie=`echo "$categorie" | sed -re 's/^[0-9]+-/'`
# Écrire le nom de la catégorie dans l'index
cat >>index.html <<EOF
<h2> $nom_categorie </h2>
EOF
```

```
for f in "$categorie/*"; do
# le nom du fichier (sans le préfixe de la catégorie)
photo=${f##*/}
# le calcul de la photo suivante
photo_suivante
# les noms des pages HTML correspondantes
PHOTO_HTML=pages_photo/${photo}/jpg/html}
SUIVANTE_HTML=${SUIVANTE}/jpg/html}
PRECEDENTE_HTML=${PRECEDENTE}/jpg/html}
# Traitement des photos et production du code HTML
```

```
done
```

```
done
```

## Structure du script principal

### Début du travail

- Le travail sur la photo `$categorie/$photo` : orienter la photo, réduire la taille, fabriquer la vignette

```
echo "Traitement de la photo $photo"
convert -auto-orient -resize 1024x768 $categorie/$photo photos/$photo
convert -auto-orient -resize 4% $categorie/$photo vignettes/$photo
```

- La production de la ligne concernant la photo dans `index.html`

```
cat >> index.html <<EOF
<a href="$PHOTO_HTML">  </a>
EOF
```

## Structure du script principal

### Traitement dans la boucle

- La production de la page HTML dédiée à la photo. Il faut faire attention à la première photo (\$PRECEDENTE vide) ...

```
# L'entête et la photo réduite
cat >> $PHOTO_HTML <<EOF
<html>
<head><title>$nom_categorie</title></head>
<body>
<h2 align="center">$nom_categorie</h2>

EOF

# Vignette photo précédente
if [[ -n $PRECEDENTE ]]; then
  cat >> $PHOTO_HTML <<EOF
  <a href="$PRECEDENTE_HTML">  </a>
EOF
fi
```

## Structure du script principal

### Traitement dans la boucle

- ... et à la dernière photo (SUIVANTE vide)

```
# Vignette photo suivante
if [[ -n $SUIVANTE ]]; then
  cat >> $PHOTO_HTML <<-EOF
  <a href="$SUIVANTE_HTML">  </a>
EOF
fi

# Le pied de page
cat >> $PHOTO_HTML <<EOF
</body>
</html>
EOF
```

- La mise à jour de la variable PRECEDENTE :

```
PRECEDENTE=$photo
```

## Et ensuite...

- Et on peaufine : regrouper les photos selon leur orientation, apprendre à utiliser des tables pour fixer le nombre de vignettes par ligne dans le fichier `index.html` et améliorer le placement des photos et des vignettes sur les pages `photo.html`, rajouter un lien oublié vers l'index dans les pages dédiées aux photos, etc.
- Et quand on est satisfait : mise sur le Web avec contrôle d'accès par `.htaccess`.  
Pour en savoir plus sur les fichiers de configuration `.htaccess` d'Apache : la page Wikipedia est très bien faite par exemple. Sinon il y a la doc Apache et des tas de tutoriels sur le sujet.

## Bugs

- Le script souffre d'un gros défaut, lequel ?

## Bugs

- Le script souffre d'un gros défaut, lequel ?
- Il ne supporte pas les espaces dans les noms des fichiers !
- Solution ?

## Bugs

- Le script souffre d'un gros défaut, lequel ?
- Il ne supporte pas les espaces dans les noms des fichiers (voir diapo 9) !
- Solution ?
- Penser à utiliser les double-quotes :

```
for f in "$categorie/*"; do  
  photo="$ {f##*/}"  
  ...  
done
```

## En résumé

On essaie de :

- ▶ se faire un plan avec des lignes très générales,
- ▶ puis affiner jusqu'à découper le travail en petites actions dont on peut facilement écrire le code correspondant
- ▶ écrire le code de chacune de ces actions
- ▶ le tester
- ▶ corriger les petites bêtises ou les oublis, etc.
- ▶ intégrer petit à petit les morceaux en testant à chaque fois

Pour la lisibilité on sépare bien l'intendance de l'action principale du script.