Manipulations de fichiers PDF

Page créée le 29 mars 2023

Manipulation de fichiers pdf en ligne de commande, par exemple pour des traitements par lots. Il existe de nombreux utilitaires sous linux dont les fonctions sont parfois identiques. Cette page présente différents exemples d'utilisation qui nous ont été utiles au fil du temps!

Le paquet poppler-utils comprend plusieurs utilitaires dont **pdfseparate, pdffonts, pdfattach, pdfunite**, etc.

pdfjam fait partie du paquet texlive-extra-utils, et permet d'utiliser les fonctions de pdfpages pour LaTeX. «*pdfjam is a shell-script front end to the LaTeX '*pdfpages*' package*»

- https://github.com/pdfjam/pdfjam
- doc pdfpages : https://mirror.ibcp.fr/pub/CTAN/macros/latex/contrib/pdfpages/pdfpages.pdf

```
pdfjam --help
```

ghostscript

- https://manpages.debian.org/bookworm/ghostscript/gs.1.en.html

Autres logiciels utiles

- **pdfposter** (https://pdfposter.readthedocs.io/en/stable/) à installer avec `sudo apt install pdfposter`
- PDFtk (basé sur java) https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/
- PDFsam (basé sur java) https://pdfsam.org/ (interface graphique, payant)
- PSPDFUtils (basé sur python) https://pypi.org/project/pspdfutils/
- masterpdfeditor https://code-industry.net/free-pdf-editor/

Extraire des feuillets d'un document pdf

Extraire toutes les pages

```
pdfseparate -f 1 document.pdf pages_%d.pdf
```

Extraire la page 3

```
pdfseparate -f 3 -l 3 document.pdf page3.pdf
```

-f (first) : première page à extraire
-l (last) : dernière page à extraire

Extraire une page et l'enregistrer en pdf dans un autre format

```
#!/bin/bash

mkdir -p output
PAGES=$(pdfinfo document.pdf | awk '/Pages:/ {print $2}')
for i in $(seq 1 $PAGES); do
    pdfjam document.pdf "$i" --papersize '{297mm,420mm}' --outfile output/page_$i.pdf
done
```

Recomposer un document pdf à partir de pages extraites d'un autre document

```
pdfjam doc1.pdf 1,3,5,7,9,11,13,15,17,19 doc2.pdf 7,11,17,19 --outfile doc_montage.pdf
```

Pour forcer en paysage

```
pdfjam doc1.pdf 1,3,5 doc2.pdf 1,9 --landscape --outfile doc_montage.pdf
```

Divers

Ajouter un fond perdu

En anglais, fond perdu = *bleed*
La commande suivante ajoute un fond perdu de 3mm sur chaque côté d'une page A5 (148x210mm), un cadre entoure la page originale (`–frame true`), les fichiers *embedded* sont conservés.

```
pdfjam --scale 1.0 --frame true --noautoscale true --papersize '{154mm,216mm}' -o document.pdf document_avec_fond_perdu.pdf
```

–frame true : cadre noir autour de la page originale

Posters

Comment découper un pdf au format A1 en 4 morceaux format A3 ?

```
# 20251013 Debian 12 @ tenko
# sudo apt install pdfposter
pdfposter -mA3 -pA1 input.pdf output.pdf
```

Agrandir un format A4 en 4 morceaux format A3

```
# UNTESTED
pdfposter -mA3 -pA4 -x2 -y2 input.pdf output.pdf
```

Passer d'un format A4 à 4xA6 sur le même document

```
pdfjam --nup 2x2 flyer.pdf flyer.pdf flyer.pdf flyer.pdf --paper a4paper --outfile flyer_podlab_4xA6.pdf
```

Remontage

assembler plusieurs pdf dans un même fichier à plusieurs pages

```
pdfjam page_1.pdf page_2.pdf page_3.pdf --paper a4paper --outfile doc.pdf
```

Passer de 12 pages A4 paysage à 6 pages A3 portraits

```
pdfjam input_A4.pdf --nup 1x2 --paper a3paper --noautoscale true --outfile output_A3.pdf
```

12 pages A4 portrait vers 6 pages A3 paysage

```
pdfjam input_A4.pdf --nup 2x1 --landscape --paper a3paper --outfile montage_A3.pdf
```

Convertir au format de papier A4

Un pdf réalisé avec convert (par exemple) ne sera pas forcèment dans un format imprimable facilement.

```
pdfjam --outfile out.pdf --paper a4paper in.pdf
pdfjam --paper a4paper --outfile out.pdf --landscape in.pdf
```

Ajouter des pages vides

```
pdfjam document.pdf '1-111,{}' -o document_complet.pdf
```

Ici, une page vide est ajoutée après la page 111 du document original, on peut moduler (par ex. '1,{},3-4,{},5-') cf. https://equa.space/notes/pdfjam/

Infos sur les polices d'un document

```
pdffonts document.pdf
```

Ces infos permettent de savoir si le fichier de fonte est intégré (*embedded*) dans le fichier pdf, son type, etc. cf. doc pdffonts

Optimiser un pdf pour réduire la taille du fichier (avec ghostscript)

```
gs -sDEVICE=pdfwrite -dCompatibilityLevel=1.4 -dEncodeColorImages=false -dNOPAUSE -dQUIET -dBATCH -sOutputFile=optimized.pdf document.pdf
```

-dEncodeColorImages=false : ne pas réencoder les images JPEG
Si on souhaite qu'aucune modification ne soit appliquée aux images on peut ajouter (source) :

```
  -dColorConversionStrategy=/LeaveColorUnchanged \
  -dEncodeColorImages=false \
  -dEncodeGrayImages=false \
  -dEncodeMonoImages=false \
```

Autre solution avec imagemagick

Lister toutes les images d'un pdf

Avec leurs caractéristiques (colorspace, width, height, x-ppi, y-ppi, etc.)

```
pdfimages -list document.pdf
```

Extraire toutes les images d'un pdf

```
pdfimages -all document.pdf /chemin/absolu/racine
```

-all : conserver les formats d'origine

Il faut absolument indiquer un chemin absolu valide!

Extraire tous les mots d'un pdf

+ Compter les occurences de chaque mot après avoir éliminé les mots de moins de 3 lettres

```
sudo apt install poppler-utils
```

frequence-mots.sh

```
pdftotext "$1" - \
  | perl -CS -ne 'while (/(\p{L}+(?:-\p{L}+)*)/g) { print lc($1), "\n"; }' \
  | awk 'length($0) >= 3' \
  | sort \
  | uniq -c \
  | sort -nr
```

Usage : `frequence-mots.sh fichier.pdf > liste_mots.txt`

Détails :
`perl -CS -ne` : CS pour travailelr en unicode (lettres accentuées, etc.), n pour traiter ligne par ligne, e pour fournir le code perl directement sur la ligne de commande
`while (/(\p{L}+(?:-\p{L}+)*)/g) { print lc($1), “\n”; }` : code perl qui se répète tant que l'expression régulière trouve des mots (les mots composés avec tiret sont considérés comme un seul mot), $1 contient le mot trouvé, lc($1) le transforme en minuscule
`awk 'length($0) >= 3'` : seuls les mots de plus de 3 lettres sont conservés
`sort` : tri par ordre alphabétique
`uniq -c` : regroupe les mots identiques sur la même ligne, préfixé par le nombre d'occurences
`sort -nr` : tri numérique des lignes en ordre décroissant

Conversion de profil de couleur

(KO par abandon)

On dirait bien que c'est un sujet complexe... Avec imagemagick on peut obtenir des informations sur le profil icc utilisé :

```
identify -format '%[colorspace]' document.pdf
```

Mais d'autres lectures indiquent que chaque élément d'un pdf peuvent avoir des profils associés différents.
Pour une conversion en CMJN, je fais un essai avec

```
gs -o document_cmjn.pdf -sDEVICE=pdfwrite -sProcessColorModel=DeviceCMYK -sColorConversionStrategy=CMYK -
sColorConversionStrategyForImages=CMYK document.pdf
```

Mais identify le détecte toujours comme du sRGB ...
En revanche avec `pdfimages -list document_cmjn.pdf` on peut voir que les images sont bien reconnues comme étant en CMJN

Test avec ghostscript et un profil ICC

```
gs -o test_fogra.pdf -sDEVICE=pdfwrite -dOverrideICC=true -sOutputICCProfile=Coated_Fogra39L_VIGC_300.icc -sColorConversionStrategy=CMYK -
sColorConversionStrategyForImages=CMYK -dProcessColorModel=/DeviceCMYK -dRenderIntent=3 -dDeviceGrayToK=true document.pdf
```

Mais ghostscript transforme toutes les images en JPEG...
nb : un pdf ne peut pas contenir d'iamges au format PNG.

Je laisse tomber pour cette fois

Reconnaissance de caractères

Avec Tesseract

Convertir un pdf en image et vice-versa

Avec Imagemagick

Ressources

Télécharger des profils de couleur (dont Fogra39) : https://www.color.org/registry/index.xalter