

Administration d'une distribution T_EX

Daniel Flipo

Ce document essaie de répondre aux questions suivantes, souvent posées sur les listes :

— j'ai bien le fichier `truc.sty` sur mon disque mais T_EX ne le voit pas, que se passe-t-il ?

— je voudrais installer des fichiers locaux (logos, classe, fichiers de macros) qui soient disponibles pour tous les utilisateurs de la machine, ou bien pour moi seulement, où dois-je les mettre ?

— j'ai installé la distribution TeX Live, que dois-je faire pour pouvoir utiliser les fontes Fourier ?

Pour répondre à ce type de questions, il faut comprendre le mécanisme de recherche des fichiers sous T_EX. Nous nous placerons dans le cas des distributions basées sur Web2C (TeX Live, macTeX, fpTeX), des indications seront données pour MikTeX/ProTeXt.

Sommaire

1	Les arborescences de base	2
2	TDS « TeX Directory Structure »	5
3	Pourquoi T _E X ne trouve-t-il pas mon fichier ?	6
4	Ajout de fichiers	8
5	Le cas de MikTeX/ProTeXt	12
6	Nouveautés	13
7	Opérations courantes de maintenance	14
8	Cohabitation de plusieurs distributions TeX sous Unix	20
9	TeX Live intégrée aux distributions Debian	21

1 Les arborescences de base

Lorsqu'on lance un exécutable de la famille T_EX (latex, bibtex, dvips, etc.) cet exécutable va lire ses variables de configuration dans un fichier `texmf.cnf`. Lequel ? la réponse est donnée par la commande `kpsewhich`, exemples :

1. TeX Live 2007 (Debian Sarge),

```
flipo% which latex
/usr/bin/latex
flipo% kpsewhich texmf.cnf
/usr/share/texmf/web2c/texmf.cnf
```

2. TeX Live 2008 sous Linux,

```
flipo% which latex
/usr/local/texlive/2008/bin/i386-linux/latex
flipo% kpsewhich texmf.cnf
/usr/local/texlive/2008/texmf.cnf
```

Extraits¹ d'un fichier `texmf.cnf` (TeX Live 2008) :

```
TEXMFDIST = $SELFAUTOPARENT/texmf-dist
TEXMFSYSVAR = $SELFAUTOPARENT/texmf-var
TEXMFLOCAL = $SELFAUTOPARENT/./texmf-local
TEXMFHOME = $HOME/texmf
TEXMF = {$TEXMFHOME,!!$TEXMFLOCAL,!!$TEXMFSYSVAR,!!$TEXMFDIST}
```

La signification de la variable `SELFAUTOPARENT` est la suivante : elle pointe sur le répertoire situé immédiatement au dessus du répertoire `bin` où sont regroupés les binaires. Ainsi, pour tout exécutable (latex par exemple) du répertoire `/usr/local/texlive/2008/bin/i386-linux/` le répertoire `TEXMFDIST` est `/usr/local/texlive/2008/texmf-dist/`, tandis que `TEXMFLOCAL` est `/usr/local/texlive/texmf-local/` (sous Unix, le symbole `..` fait remonter d'un niveau dans l'arborescence).

Rappelons que sous Unix, les noms des variables d'environnement sont écrits en capitales et qu'on fait précéder leur nom d'un `$` pour désigner leur valeur. Ainsi `$HOME`, valeur de la variable `HOME`, désigne le répertoire personnel de chaque utilisateur, par exemple `/home/dupont` pour l'utilisateur `dupont`.

La valeur donnée ci-dessus à `TEXMF` impose à T_EX de parcourir successivement les arborescences `$TEXMFHOME`, puis `$TEXMFLOCAL`, puis `$TEXMFSYSVAR` et enfin `$TEXMFDIST`, c'est-à-dire d'abord l'arborescence personnelle de l'utilisateur, puis

1. La valeur de `TEXMF` a été simplifiée pour alléger l'exposé, nous reviendrons plus loin (voir section 6) sur la liste complète des répertoires contenus dans `TEXMF`.

l'arborescence gérée par l'administrateur de la machine et enfin les répertoires installés par la distribution T_EX. On arrête la recherche dès qu'un fichier correspondant au nom donné est trouvé, ainsi un fichier `truc.sty` situé dans `$TEXMFHOME` masquera complètement les fichiers de même nom contenus dans les autres arborescences.

Il reste à expliquer la signification des « !! ». Il est courant qu'une arborescence T_EX contienne plusieurs milliers de fichiers, aussi la recherche d'un fichier sur le disque prendrait trop de temps, il est bien plus efficace de constituer une base de données qui répertorie la position de tous les fichiers de l'arborescence et de ne consulter *que* la base de données pour localiser le fichier. La présence du signe !! devant le nom d'un répertoire ordonne à T_EX de se limiter à parcourir la base de données ; si le nom du fichier n'y figure pas, on admet que celui-ci n'est pas sur le disque et on le cherche alors dans l'arborescence suivante.

L'avantage évident est d'accélérer notablement la recherche, la contrainte est qu'il est impératif de tenir à jour les bases de données, faute de quoi des fichiers physiquement présents sur le disque ne seront pas vus.

Ces bases de données sont créées grâce à la commande `ls -R` d'Unix, d'où leur nom `ls-R`, ce sont des fichiers placés à la racine de l'arborescence qu'ils décrivent, par exemple `/usr/local/texlive/texmf-local/ls-R` pour l'arborescence locale.

Dans l'exemple ci-dessus, on a fait le choix, souvent judicieux, de limiter la recherche aux fichiers `ls-R`, sauf pour l'arborescence personnelle `$TEXMFHOME` de chaque utilisateur (la seule de la liste à ne pas être précédée de !!). En effet celle-ci ne contient en général que quelques dizaines de fichiers, la recherche sur le disque n'est pas un problème dans ce cas. De plus, on peut craindre qu'un utilisateur non averti oublie de mettre à jour son fichier `ls-R` après avoir ajouté un fichier...

Il n'y a aucun intérêt à ajouter un fichier `ls-R` dans `$TEXMFHOME` si son nom n'est pas précédé de !! dans la liste TEXMF. En effet, un fichier `ls-R` accélérerait (un tout petit peu) la recherche des fichiers présents dans cette arborescence, mais retarderait celle des autres fichiers (beaucoup plus nombreux) puisque T_EX les chercherait d'abord (sans succès) sur le disque dans l'arborescence `$TEXMFHOME` avant de passer aux arborescences suivantes.

Poursuivons la lecture du fichier `texmf.cnf`, on y trouve des lignes du genre

```
TEXFORMATS = .;$TEXMF/web2c
TEXINPUTS.latex = .;$TEXMF/tex/{latex,generic,}//
TEXINPUTS.pdftex = .;$TEXMF/{pdftex,tex,}/{plain,generic,}//
TEXINPUTS.context = .;$TEXMF/tex/{context,plain,generic,}//
VFFONTS = .;$TEXMF/fonts/vf//
TFMFonts = .;{$TEXMF/fonts,$VARTEXFONTS}/tfm//
```

```
T1FONTS = .;$TEXMF/fonts/type1//
TEXFONTPMAPS = .;$TEXMF/fonts/map/{pdftex,dvips,dvipdfm}//
BIBINPUTS = .;$TEXMF/bibtex/bib//
BSTINPUTS = .;$TEXMF/bibtex/{bst,csf}//
```

Les fichiers sont rangés selon leur type dans une partie spécifique de l'arborescence TEXMF. Ceci contribue à accélérer la recherche.

Le « . » désigne le répertoire courant, il est en général en tête du chemin de recherche, ce qui signifie que les fichiers présents dans le répertoire courant sont toujours vus en priorité.

Les « ; » séparent les répertoires qui seront parcourus successivement.

La variable TEXFORMATS indique où doivent se trouver les formats (fichiers `latex.fmt`, `pdflatex.fmt`, `context.fmt`, etc.) : le répertoire courant, ou à défaut le sous-répertoire `web2c` de l'un des répertoires dont la liste est donnée par le contenu de la variable TEXMF.

Il est possible de suffixer le nom d'une variable par un nom de format ou de programme. La variable TEXINPUTS fait largement usage de cette possibilité pour différencier les chemins de recherche selon le format utilisé. Ainsi, le format `latex` cherchera ses fichiers dans le répertoire courant, puis dans le sous-répertoire `tex/latex` puis dans `tex/generic`² de chacun des répertoires listés dans TEXMF. Le « // » à la fin indique que la recherche doit avoir lieu non seulement dans les répertoires `tex/latex` et `tex/generic` mais aussi dans tous leurs sous-répertoires.

Les valeurs de `TEXINPUTS.pdftex` et `TEXINPUTS.context` s'interprètent de la même façon. Le répertoire `generic` est commun à tous les formats. Ainsi, un utilisateur qui souhaiterait pouvoir utiliser ses logos à la fois avec \LaTeX et ConTeXt, sans devoir les recopier dans le répertoire courant à chaque fois, pourrait créer une chaîne de répertoires `texmf/tex/generic/logos` à la racine de son répertoire personnel et y ranger tous ses logos. En revanche, des fichiers mis directement dans `texmf` ou dans `texmf/logos` *ne seraient vus par aucun format*.

Les autres lignes extraites du fichier `texmf.cnf` obéissent à la même syntaxe, elles indiquent où doivent se trouver les fichiers suffixe `.vf` (VFFONTS), `.tfm` (TFM FONTS), `.pfa` ou `.pfb` (T1FONTS), `.map` (TEXFONTPMAPS), etc.

Ce qui précède devrait permettre de comprendre la première partie du fichier `texmf.cnf` (celle relative aux chemins de recherche). Récapitulons les points principaux développés ci-dessus :

— Plusieurs arborescences peuvent être parcourues successivement (personnelle, locale, distribution, ...).

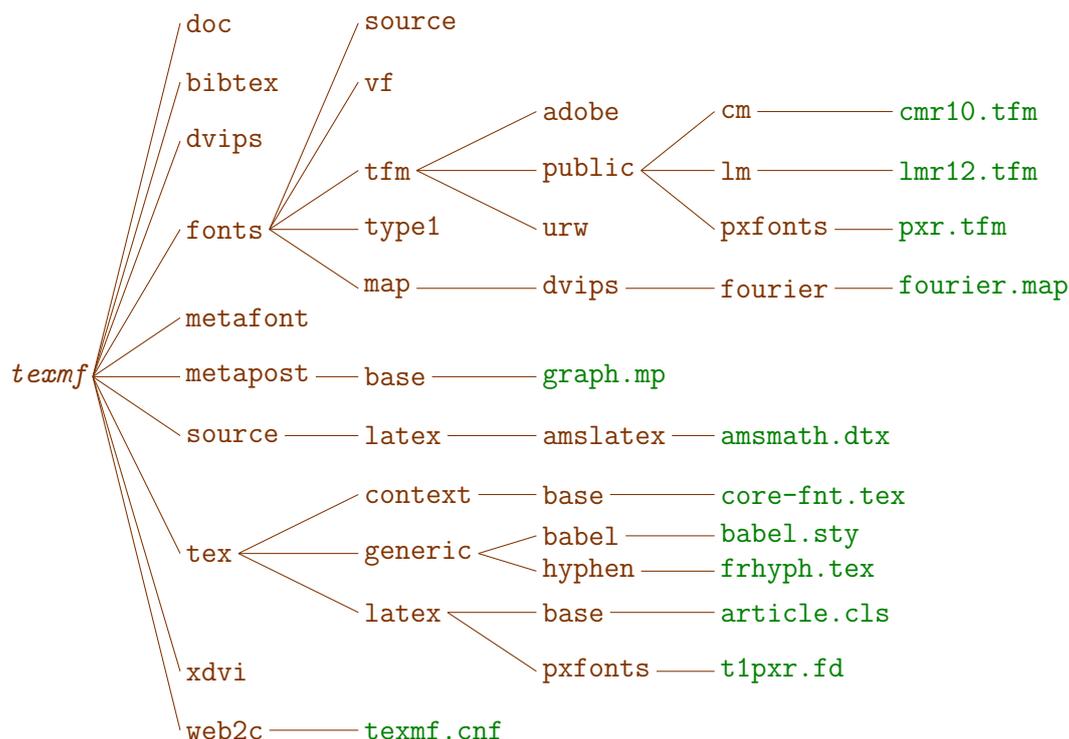
2. Les accolades encadrent la liste des répertoires de même niveau à parcourir successivement.

- La recherche prend en compte la base de données `ls-R` en priorité ou exclusivement (directive `!!`), car scruter le disque est inefficace.
- La recherche n'a lieu que dans une partie de l'arborescence, partie qui peut dépendre du programme utilisé. Ceci impose que chaque arborescence respecte une architecture standard dite TDS « TeX Directory Structure » que nous allons décrire maintenant.

2 TDS « TeX Directory Structure »

La TDS est une recommandation d'architecture pour le rangement des fichiers utilisés par \TeX et consorts. Elle émane d'un groupe de travail du TUG et a été adoptée par toutes les distributions actuelles.

La version actuelle (1.1) date de juin 2004. Les spécifications ont été traduites en français par Jean-Côme Charpentier dans le numéro 44-45 des Cahiers GUTenberg paru fin 2004. L'article, dont je recommande la lecture, détaille les choix faits et les raisons qui les ont motivés. Le schéma ci-dessous devrait permettre au lecteur de se faire une idée de l'architecture TDS.



Au premier niveau on a essentiellement une répartition par *programmes* (bibtex,

dvips, metafont, metapost, tex, xdvi), complétée par des répertoires génériques comme doc, fonts, source et web2c.

Considérons le répertoire fonts. Le sous-répertoire source contient les fichiers sources METAFONT (de suffixe .mf) pour les polices disponibles dans ce format, comme les « cm » ou les « ec ». Ne pas confondre ce répertoire avec le répertoire *texmf/metafont*³ qui contient lui les fichiers de base pour le programme METAFONT. Les sous-répertoires source, vf, et type1 ont la même structure que tfm : un premier niveau désigne le fondeur (adobe, urw par exemple ou public), le second désigne le nom de la famille de fontes.

Le répertoire tex est subdivisé selon les formats context, latex par exemple et generic pour ce qui est commun à plusieurs formats : babel qui sert à latex mais aussi à plain est dans generic. Les fichiers de césures sont regroupés sous *texmf/tex/generic/hyphen*. Le sous-répertoire *texmf/tex/latex* contient plusieurs centaines de sous-répertoires (un par extension!).

En cas de doute sur l'endroit où doit se trouver tel ou tel fichier, le mieux est de se reporter à la première partie du fichier *texmf.cnf* qui explicite la façon dont l'auteur de la distribution a mis en œuvre la TDS.

3 Pourquoi T_EX ne trouve-t-il pas mon fichier ?

Nous sommes maintenant en mesure de répondre à cette question. Il n'y a que deux raisons (non exclusives l'une de l'autre) qui puissent expliquer que T_EX ne voie pas un fichier présent sur le disque :

1. le fichier n'est pas au bon endroit dans l'arborescence,
2. le fichier se trouve dans une arborescence dont le nom est précédé de !! dans la variable TEXMF mais il ne figure pas dans la base de données ls-R correspondante (fichier ajouté sur le disque sans reconstruction de la base).

Fichier mal placé

Rappelons que la recherche n'a lieu que *dans une partie* de chaque arborescence *texmf*, partie qui dépend du type de fichier recherché. Il faut s'assurer que le fichier est bien dans un répertoire adéquat en consultant la TDS ou le fichier *texmf.cnf*, notamment la valeur des variables TEXINPUTS et consorts (voir p. 3). Des exemples indiquant où placer les fichiers selon leur type seront donnés p. 9.

3. la notation *texmf* désigne l'un quelconque des répertoires listés dans la variable TEXMF.

Exemples d'erreurs classiques :

— un utilisateur a mis son fichier de macros personnelles pour \LaTeX sous $\$HOME$ ou sous $\$HOME/texmf$ au lieu de le mettre sous $\$HOME/texmf/tex/latex$ (ou dans un sous-répertoire de ce dernier) ;

— l'administrateur d'une machine a mis une classe locale directement sous $\$TEXMFLOCAL$ ou bien sous $\$TEXMFLOCAL/tex$ au lieu de la mettre sous $\$TEXMFLOCAL/tex/latex$;

— de même, mettre tous les fichiers contenus dans l'extension Fourier en vrac dans $\$TEXMFLOCAL/fourier$ ou même sous $\$TEXMFLOCAL/fonts/fourier$ n'a aucune chance de marcher.

Base $ls-R$ pas à jour

Après ajout d'un ou plusieurs fichiers dans une arborescence *texmf*, la mise à jour du fichier $ls-R$ correspondant s'impose. Elle peut se faire en lançant la commande `mktexlsr` dans une fenêtre « Terminal » ou à l'aide de l'utilitaire `texconfig` (option `rehash`). Normalement, il faut avoir les droits de « root » pour pouvoir modifier les fichiers $ls-R$, c'est donc « root » qui doit lancer l'une de ces commandes. Tout se passe sans problème si « root » et les utilisateurs non privilégiés utilisent la même distribution \TeX , c'est le cas en général, sauf lorsqu'on installe une distribution TeX Live plus récente, par exemple TeX Live 2008, en plus de la distribution intégrée (TeX Live 2007 sous Debian Sarge).

En effet, les distributions Linux font de \TeX un composant obligatoire, il est nécessaire que les binaires de la famille \TeX vus par « root » soient ceux de la distribution, même si les autres utilisateurs font appel à des versions plus récentes (TeX Live 2008 plutôt que TeX Live 2007 inclus dans Debian Sarge par exemple). Pour mettre à jour les fichiers $ls-R$ de la distribution \TeX installée en plus de la distribution officielle sur une machine Linux, on a le choix entre deux solutions :

1. soit on lance sous « root » la commande `mktexlsr` ou `texconfig` *en donnant le chemin complet*, par exemple

```
root# /usr/local/texlive/2008/bin/i386-linux/mktexlsr
```

2. soit on souhaite qu'un groupe d'utilisateurs (avertis) puisse mettre à jour les fichiers $ls-R$ de TeX Live, et dans ce cas on définit un groupe spécifique, par exemple « tex », on ajoute les utilisateurs autorisés à ce groupe et on donne des droits en écriture au groupe « tex » sur les fichiers $ls-R$.

Exemple : la distribution TeX Live est sous `/usr/local/texlive`, les utilisateurs « dupont » et « durant » doivent être habilités à mettre à jour sous leur nom les fichiers $ls-R$ de TeX Live, voici un moyen de le faire (sous « root ») :

```
root# addgroup tex
```

```
root# adduser dupont tex
root# adduser durant tex
root# find /usr/local/texlive -name 'ls-R' -exec chgrp tex {} \;
root# find /usr/local/texlive -name 'ls-R' -exec chmod 664 {} \;
```

Ensuite il suffira aux utilisateurs « dupont » ou « durant » de lancer la commande `mktexlsr` depuis leur compte pour mettre à jour les fichiers `ls-R` de TeX Live.

Si « root » lance la commande `mktexlsr`, il mettra à jour les fichiers `ls-R` de la distribution T_EX officielle (TeX Live 2007 sous Debian Sarge), *pas ceux* de la distribution supplémentaire (TeX Live 2008) ! Nous reviendrons à la section 8 sur la cohabitation de plusieurs distributions T_EX sur une même machine.

Dans tous les cas, lors du lancement des commandes `mktexlsr` ou `texconfig`, les noms complets des fichiers `ls-R` mis à jour sont affichés, ce qui permet de savoir si la commande a opéré sur les bons fichiers.

4 Ajout de fichiers

Lors de l'installation d'une nouvelle extension, d'une nouvelle classe ou d'une nouvelle fonte, il y a deux types de choix à faire :

1. décider dans quelle arborescence (personnelle, locale, distribution) il est préférable de faire l'installation ;
2. décider dans quel répertoire de cette arborescence faire l'installation.

Dans quelle arborescence ?

Le choix de l'arborescence est assez facile à faire en général. Prenons tout d'abord le cas de la distribution TeX Live.

— S'il s'agit d'une mise à jour de fichiers existant déjà dans la distribution standard, il faut évidemment mettre les nouvelles versions en lieu et place des anciennes, dans l'arborescence `TEXMFDIST` donc.

— S'il s'agit de nouveaux fichiers (nouvelle extension, nouvelle fonte) apparus sur CTAN *sans restriction de diffusion particulière*, il y a tout lieu de penser qu'ils seront intégrés à la prochaine distribution TeX Live et donc les mettre dans `TEXMFDIST` est un bon choix a priori. C'est ce que j'ai fait par exemple lorsque la classe `beamer` est sortie. Elle a effectivement fait partie de TeX Live l'année suivante, dans une version largement améliorée ; si j'avais fait l'installation initiale dans `TEXMFLOCAL`, j'aurais dû la désinstaller l'année suivante pour bénéficier de la

version améliorée intégrée à TeX Live (rappelons que les fichiers de TEXMFLOCAL sont prioritaires sur ceux de TEXMFDIST).

En revanche des fichiers dont les licences ne sont ni LPPL ni GPL risquent fort d'être écartés de TeX Live, dans ce cas les mettre dans TEXMFLOCAL est préférable.

— S'il s'agit d'extensions personnelles ou communiquées par des collègues à titre privé, le choix de TEXMFLOCAL ou de TEXMFHOME s'impose selon que tous les utilisateurs de la machine peuvent y avoir accès ou non.

— S'il s'agit d'une fonte commerciale, le choix de TEXMFDIST est exclu (sauf à être prêt à refaire l'installation tous les ans après celle de TeX Live!) ; la licence imposera l'installation soit dans TEXMFHOME (licence mono-utilisateur) soit dans TEXMFLOCAL (licence multi-utilisateurs).

Considérons maintenant le cas d'une distribution T_EX installée de base avec Linux, le choix est encore plus simple : *aucune* modification ou ajout ne doit être faite à la main dans la distribution Linux. Il *faut* passer par les outils standard (apt-get ou aptitude sous Debian, rpm ou yum sous RedHat, etc.) pour mettre à jour ou ajouter des paquets. Ceci signifie qu'il faut attendre la sortie des mises à jour sous forme de paquets .deb ou .rpm, ce qui peut être assez frustrant : sous Debian stable, par exemple beaucoup de fichiers sont obsolètes...

Des ajouts peuvent être faits comme avec TeX Live dans les arborescences TEXMFLOCAL ou TEXMFHOME selon les mêmes critères.

Où dans l'arborescence ?

La seconde décision concerne la localisation des fichiers à l'intérieur de l'arborescence choisie. Ce choix doit être conforme à la TDS ou plus précisément cohérent avec les valeurs des variables figurant dans le fichier `texmf.cnf` : voir section 1 p. 3, variables TEXINPUTS, VFFONTS, etc.

En général, le suffixe suffit à décider de l'endroit où mettre le fichier, exemples :

— les fichiers .cls (classes L^AT_EX), .sty (extensions), .cfg (fichiers de configuration), .fd (définition de fonte), etc., vont dans (un sous-répertoire de) `texmf/tex/latex/` ;

— les images à inclure dans des fichiers L^AT_EX ou pdfL^AT_EX, de suffixe .eps, .pdf, .png ou .jpg ainsi que les fichiers de suffixe numérique produits par METAPOST vont soit dans (un sous-répertoire de) `texmf/tex/latex/`, soit dans (un sous-répertoire de) `texmf/tex/generic/` si elles sont susceptibles d'être aussi utilisées avec d'autres formats comme ConT_EXt ou plainT_EX ;

— les fichiers source METAPOST de suffixe .mp vont eux dans (un sous-répertoire de) `texmf/metapost/`

— les fichiers .dtx, .ins, etc. vont dans (un sous-répertoire de) `texmf/source/latex/` ; ils ne sont jamais utilisés directement, ils servent à fabriquer les

fichiers utilisables `.cls`, `.sty`, etc. (voir p. 10) ;

— les fichiers de documentation associés à une extension ou une classe \LaTeX , en `.pdf` ou `.dvi` ainsi que les fichiers source `.tex` qui servent à les créer vont dans (un sous-répertoire de) `texmf/doc/latex/` ;

— les fichiers `.tfm` (fichiers de métriques des fontes) vont dans `texmf/fonts/tfm/fondeur/famille/` ;

— les fichiers `.vf` (fichiers de fontes virtuelles) vont dans `texmf/fonts/vf/fondeur/famille/` ;

— les fichiers `.pfa` et `.pfb` (fichiers de glyphes des fontes type 1) vont dans `texmf/fonts/type1/fondeur/famille/` ;

— les fichiers `.map` associés aux fontes type 1 vont dans `texmf/fonts/map/dvips/` ;

— les fichiers de documentation associés à une fonte, en `.pdf` ou `.dvi` ainsi que les fichiers source `.tex` qui servent à les créer vont dans (un sous-répertoire de) `texmf/doc/fonts/` ;

Il convient de respecter une règle supplémentaire selon laquelle tout répertoire ne doit contenir que des fichiers ou que des répertoires, jamais un mélange des deux. Le plus simple est de créer un sous-répertoire pour chaque extension, c'est ce qui amène, par exemple, à éclater les fichiers de l'extension `tabvar` en 8 sous-répertoires :

`texmf/doc/latex/tabvar/` (README, `tabvar.pdf`, `demo.pdf`, `demo.tex`)

`texmf/source/latex/tabvar/` (`tabvar.dtx`, `tabvar.ins`)

`texmf/tex/latex/tabvar/` (`tabvar.sty`, `tabvar.cfg`, `tabvar.[123]`)

`texmf/metapost/tabvar/` (`tabvar.mp`)

`texmf/fonts/afm/public/tabvar/` (`tabvar.afm`)

`texmf/fonts/tfm/public/tabvar/` (`tabvar.tfm`)

`texmf/fonts/type1/public/tabvar/` (`tabvar.pfb`)

`texmf/fonts/map/dvips/tabvar/` (`tabvar.map`)

dont les 5 derniers ne contiennent qu'un seul fichier. L'autre solution consisterait à mettre, par exemple, tous les fichiers `.tfm` des 125 familles qui font partie de « public » en vrac dans `texmf/fonts/tfm/public/` !

Ne pas oublier de mettre à jour les bases de données `ls-R` après avoir mis en place les fichiers (commandes `mktexlsr` ou `texconfig`, voir section 3).

Cas d'une extension \LaTeX fournie en `.dtx`

Une extension \LaTeX récupérée sur CTAN peut se présenter sous deux formes : soit un ensemble de fichiers prêts à l'emploi (suffixes `.cls`, `.sty`, `.cfg`, `.fd`,

etc.), soit sous forme de « fichiers de code source documenté » de suffixe `.dtx`, accompagnés des fichiers d'installation de suffixe `.ins`. Le premier cas a été traité ci-dessus.

Dans le second cas, il faut d'abord extraire les fichiers utiles avant de les installer au bon endroit dans une arborescence *texmf*, les fichiers `.dtx` n'étant pas directement utilisables par \LaTeX . Voici la marche à suivre : on crée un répertoire du nom de l'extension dans *texmf/source/latex* et on y met les fichiers `.dtx` et `.ins`.

L'extraction se fait en compilant le ou les fichiers de suffixe `.ins` avec \LaTeX . En principe, un message final donne la liste des fichiers « à déplacer dans un répertoire visible par \LaTeX », c'est normalement *texmf/tex/latex/extension*.

L'extraction de la documentation se fait en compilant le ou les fichiers de suffixe `.dtx`, de préférence avec `pdf \LaTeX` . On déplace ensuite les fichiers `.pdf` produits dans *texmf/doc/latex/extension* et on termine par la mise à jour des bases de données `ls-R`.

Cas d'une fonte type 1

Lorsqu'on achète une police commerciale type 1, le vendeur fournit les glyphes sous la forme de fichiers `.pfb` et les métriques sous forme de fichiers `.afm`. Pour pouvoir utiliser la police avec \LaTeX , il manque les fichiers `.tfm` (métriques au format \TeX), les fichiers `.vf` (polices virtuelles servant au recodage des caractères, la police fournie étant en général au codage Adobe différent de celui qu'utilise \TeX), un fichier `.map` (qui indique à \TeX où aller chercher les glyphes), des fichiers `.fd` (définition des polices) et éventuellement un fichier `.sty` pour faciliter l'utilisation de la police.

L'outil permettant de créer ces fichiers s'appelle `fontinst`, sa description sort du cadre du présent article⁴. Sur l'utilisation de `fontinst` on pourra consulter la documentation qui l'accompagne (fichiers `fontinst.pdf` et `et99-font-tutorial.pdf`) ou le livre de Bernard Desgraupes [5].

Heureusement, le travail d'adaptation à \TeX des polices libres les plus courantes a déjà été fait : l'extension `psnfss` qui est installée de base dans toutes les distributions donne accès aux polices AvantGarde, Bookman, Zapf Chancery, Charter, Courier, Helvetica, NewCenturySchoolBook, Palatino et Times.

Le cas de la police Utopia qui sert de base à la famille Fourier est spécial : Adobe fournit gratuitement les 4 fichiers `putr8a.pfb` (Utopia-Regular), `putri8a.pfb` (Utopia-Italic), `putb8a.pfb` (Utopia-Bold) et `putbi8a.pfb` (Utopia-BoldItalic),

4. Désormais, les polices commerciales sont vendues au format OpenType, ce qui complique encore un peu plus leur utilisation avec \LaTeX . Pour utiliser des fontes OpenType le passage à \XeLaTeX s'impose!

mais leur licence d'utilisation n'est pas considérée comme « suffisamment libre » pour être intégrée dans TeXLive ou dans les distributions Debian.

Dans la distribution TeXLive tout est prêt pour l'utilisation d'Utopia et de Fourier, seuls manquent les 4 fichiers (indispensables !) `put*8a.pfb` listés ci-dessus. Pour les installer, procéder comme suit :

1. Récupérer ces 4 fichiers sur CTAN dans `tex-archive/fonts/utopia/` et les mettre dans l'arborescence TEXMFLOCAL dans le répertoire `texmf/fonts/type1/adobe/utopia/` ; on peut aussi récupérer les 4 fichiers `.afm` et les mettre dans `texmf/fonts/afm/adobe/utopia/` mais \LaTeX ne s'en sert pas.

2. Mettre à jour la base de données `ls-R` (commande `mktexlsr` voir section 3).

3. Il reste à faire savoir aux pilotes de sortie (`dvips`, `pdftex`, `dvipdfm`), comment utiliser ces polices type 1 directement sans création de polices bitmaps. Ceci nécessite d'ajouter pour chaque police une ligne dans les fichiers `psfonts.map` (pour `dvips`), `pdftex.map` (pour `pdftex`) et `dvipdfm.map` (pour `dvipdfm`).

La distributions TeXLive fournit des scripts, `updmap` et `updmap-sys`⁵, permettant d'automatiser la création et la mise à jour de ces fichiers `.map`. La commande à utiliser est

```
root# updmap-sys --enable Map fourier.map
```

Il faut évidemment s'assurer d'utiliser le script *de la distribution TeXLive* en donnant éventuellement le chemin complet (voir ce qui a été dit sur `mktexlsr` à la fin de la section 3).

5 Le cas de MikTeX/ProTeXt

Par défaut, les répertoires `texmf` sont `C:/texmf` (distribution de base) et `C:/localtexmf` (pour les ajouts locaux). Ils respectent également l'architecture TDS et leur contenu est mémorisé dans des fichiers FNDB (équivalents des `ls-R`). On accède au programme de mise à jour de ces fichiers dans les options de MikTeX (Menu Démarrer → Tous les programmes → Miktex → Miktex Options).

Les fichiers de configuration propres à MikTeX sont dans `C:/texmf/miktex/config` (au lieu de `web2c`), l'équivalent de `texmf.cnf` s'appelle `miktex.ini`.

Le fichier `updmap.cfg` se trouve lui dans `C:/texmf/web2c`. L'équivalent de la commande `updmap` est `initexmf --mkmaps`, à lancer dans une fenêtre de commandes.

Il n'y a pas d'équivalent à la commande `kpsewhich`.

5. Le script `updmap-sys` met à jour les fichiers `.map` communs à tous les utilisateurs, tandis que `updmap` ne modifie que ceux de l'utilisateur (dans `TEXMFHOME`).

La documentation complète sur MikTeX se trouve dans le fichier `C:/texmf/doc/miktex/miktex.pdf`.

6 Nouveautés

Les scripts `updmap`, `fmtutil` et `texconfig` opèrent *au niveau de l'utilisateur* : ils créent les fichiers `.map` et les formats dans le répertoire personnel de l'utilisateur qui les lance.

De nouveaux scripts `updmap-sys`, `fmtutil-sys` et `texconfig-sys` ont été ajoutés pour la configuration *au niveau de la machine* : ils créent les fichiers `.map` et les formats utilisables par tous les utilisateurs de la machine. Il faut avoir les droits de « *root* » pour les lancer. Ces scripts remplacent les `updmap`, `fmtutil` et `texconfig` des versions antérieures de TeX Live.

Dans l'exemple donné au début de la section 1, nous avons volontairement tronqué la liste (variable `TEXMF`) des arborescences où les fichiers sont recherchés. De nouvelles arborescences ont été introduites, certaines ont été renommées. En voici la liste complète.

Au niveau « *distribution* », c'est-à-dire ce qui est contenu dans la distribution standard TeX Live, il y a maintenant quatre arborescences :

`TEXMFDIST` contient les fichiers de macros pour les différents formats (Plain, \LaTeX , ConTeXt, etc.), les fontes, etc. Rien ne devrait être ni ajouté ni supprimé dans cette arborescence.

`TEXMFMAIN` contient les éléments dépendant de la machine, comme les scripts (`web2c/mktexdir`, etc.), le fichier de configuration principal (`web2c/texmf.cnf`), les fichiers `.pool`, etc.

`TEXMFSYSCONFIG` c'est là que les scripts `updmap-sys` et `fmtutil-sys` recherchent leurs fichiers de configuration en priorité. `texconfig-sys` y place les versions modifiées des fichiers de configuration.

`TEXMFSYSVAR`⁶ c'est là que les scripts `texconfig-sys`, `updmap-sys` et `fmtutil-sys` rangent les formats et les fichiers `.map` qu'ils produisent.

Au niveau « *local* » il y a l'arborescence `TEXMFLOCAL`, c'est là que l'administrateur de la machine place les classes et extensions locales, les fontes non disponibles dans la distribution standard, etc.

Au niveau « *personnel* » il y a maintenant trois arborescences :

6. Elle s'appelait `VARTEXMF` dans les distributions antérieures et a été renommée par souci d'homogénéité.

7. Elle s'appelait `HOMETEXMF` dans les distributions antérieures et a été renommée par souci d'homogénéité.

`TEXMFHOME`⁷ contient les fichiers personnels (extensions, classes, fontes) de l'utilisateur.

`TEXMFCONFIG` c'est là que les scripts `updmap` et `fmtutil` recherchent leurs fichiers de configuration en priorité. `texconfig` y place les versions modifiées des fichiers de configuration.

`TEXMFVAR` c'est là que les scripts `texconfig`, `updmap` et `fmtutil` rangent les formats et les fichiers `.map` qu'ils produisent.

La valeur standard de la variable `TEXMF` dans TeX Live 2008 est

```
TEXMF ={$TEXMFCONFIG,$TEXMFVAR,$TEXMFHOME,  
        !!$TEXMFSYSCONFIG,!!$TEXMFSYSVAR,  
        !!$TEXMFMAIN,!!$TEXMFLOCAL,!!$TEXMFDIST}
```

7 Opérations courantes de maintenance

Les scripts `texconfig` et `texconfig-sys` présentés ci-dessus (voir section 6) offrent une interface graphique pour gérer un certain nombre de problèmes de configuration et de maintenance. Personnellement, je préfère éditer directement les fichiers de configuration, c'est la méthode que je vais décrire dans cette section.

Si on n'utilise pas les scripts `texconfig*` pour modifier la configuration d'une distribution TeX, la première chose à faire est de trouver le nom du fichier à modifier (`language.dat`, `fmtutil.cnf`, `mktex.cnf`, `updmap.cfg`, `texmf.cnf`, etc., leurs rôles respectifs sont décrits ci-dessous) et s'il y en a plusieurs du même nom, ce qui est fréquent, déterminer lequel est pris en compte. Ceci se fait grâce à la commande `kpsewhich : kpsewhich language.dat` retourne le chemin complet du fichier `language.dat` pris en compte.

Ensuite deux approches sont possibles :

— soit on modifie le fichier actif⁸ mais dans ce cas les modifications ne survivront pas à un changement de version (passage de TeX Live 2007 à TeX Live 2008 par exemple),

— soit on copie le fichier actif dans l'arborescence `TEXMFLOCAL` (sous le même nom et à la même position relative) et on modifie cette version, ce qui a l'avantage de regrouper tout ce qui fait partie de la configuration locale en dehors de l'arborescence principale.

Ceux qui choisissent la seconde approche devront prendre garde au fait que les scripts `updmap-sys` et `fmtutil-sys` cherchent leur fichier de configuration

8. C'est ce qui se passe lorsqu'on utilise le script `texconfig-sys` avec TeX Live 2008 ou `texconfig` avec les distributions TeX Live antérieures.

(`updmap.cfg` ou `fmtutil.cnf`) en priorité dans l'arborescence `TEXMFSYSCONFIG` qui est parcourue *avant* `TEXMFLOCAL` (voir la valeur standard de `TEXMF` p. 14). Une solution, pour que les fichiers `updmap.cfg` et `fmtutil.cnf` de `TEXMFLOCAL` soient pris en compte, est de définir `TEXMFSYSCONFIG` comme `TEXMFLOCAL` dans le fichier `texmf.cnf`.

Le fichier `texmf.cnf` est normalement dans `TEXMFMAIN`. Pour regrouper dans `TEXMFLOCAL` *toutes* les modifications de configuration, on peut procéder de la manière suivante :

— définir la variable d'environnement `TEXMFCNF` comme une suite de chemins, séparés par un « : », avec en tête le répertoire local et ensuite le répertoire standard où se trouvent les fichiers `texmf.cnf` à prendre en compte, par exemple

```
export TEXMFCNF=/usr/local/texlive/texmf-local/web2c:  
                /usr/local/texlive/2008/texmf/web2c
```

Cette syntaxe est valable pour un *shell* `bash`⁹, cette déclaration est à placer dans le fichier `.bash_profile` de chaque utilisateur pour être exécutée à chaque connexion.

Pour un *shell* `csh`, remplacer « `export TEXMFCNF=` » par « `setenv TEXMFCNF` » et mettre cette déclaration dans le fichier `.cshrc`.

— redéfinir, dans le fichier `texmf.cnf` local, uniquement les variables dont les valeurs ne conviennent pas, par exemple

```
TEXMFLOCAL = /usr/local/texlive/texmf-local  
TEXMFSYSCONFIG = $TEXMFLOCAL  
TEXMF = {!!$TEXMFCNF,!!$TEXMFVAR,$TEXMFHOME,  
        !!$TEXMFSYSCONFIG,  
        !!$TEXMFSYSVAR,!!$TEXMFMAIN,!!$TEXMFDIST}
```

Noter que `TEXMFLOCAL` a été supprimé de la liste, cette arborescence étant déjà prise en compte en tant que `TEXMFSYSCONFIG`.

Ainsi le fichier `/usr/local/texlive/texmf-local/web2c/texmf.cnf` est lu en premier, les valeurs des variables affectées dans ce fichier ne seront pas remises en cause lors de la lecture du fichier standard `/usr/local/texlive/2008/texmf/web2c/texmf.cnf` (ceci est une propriété de `kpathsea`) ; celui-ci donnera aux variables non définies précédemment les valeurs prévues par TeX Live 2008.

Formats : le fichier `fmtutil.cnf`

Le fichier `fmtutil.cnf` contient la liste des formats disponibles et la façon de les fabriquer.

Le format des lignes est facile à comprendre, par exemple la ligne

9. Le nom du *shell* utilisé s'obtient par la commande `echo $SHELL`.

```
latex pdfetex language.dat -translate-file=cp227.tcx *latex.ini
```

créé le format `latex.fmt` (lancé par la commande `latex`). Le moteur utilisé est `pdfetex`¹⁰, il fait appel au fichier `language.dat` pour les césures et au fichier `cp227.tcx` pour la traduction des caractères en sortie¹¹, les commandes de création du format sont dans le fichier `latex.ini`.

La création des formats se fait grâce à la commande `fmtutil` ou `fmtutil-sys` (voir section 6) :

```
fmtutil-sys --all
```

 crée ou régénère tous les formats décrits dans le fichier `fmtutil.cnf` (sauf ceux dont le nom est précédé de `#!`),

```
fmtutil-sys --byfmt latex
```

 crée ou régénère le format `latex.fmt`,

```
fmtutil-sys --help
```

 fournit les autres options disponibles.

Césures : le fichier `language.dat`

Les endroits où (L)T_EX est autorisé à couper un mot en bout de ligne sont déterminés par l'inclusion *dans le format* d'une liste de motifs de césures. Cette liste dépend évidemment de la langue de travail (on peut couper `si-gnal` en français, mais en c'est anglais `sig-nal`).

Le fichier `language.dat` contient les noms des langues disponibles (pour `babel` en particulier). C'est dans ce fichier qu'on peut ajouter des langues ou choisir le fichier de motifs à prendre en compte pour une langue donnée.

Tous les formats ne font pas appel à `language.dat` (`tex` par exemple ne connaît que l'américain), voir le fichier `fmtutil.cnf` décrit ci-dessus.

La lecture du fichier `language.dat` est intéressante. Les noms de langues sont ceux qu'on donne comme option à l'appel de l'extension `babel`. On y voit notamment que le nom `english` et ses alias `usenglish` ou `american` correspondent à l'américain (fichier `hyphen.tex`). Pour accéder à l'anglais britannique il faut donner à `babel` l'option `british` ou `ukenglish` (fichier `ukhyphen.tex`).

Les changements faits dans le fichier `language.dat` ne prennent effet qu'après recompilation du format, c'est l'opération décrite dans la sous-section précédente.

10. La majorité des formats n'utilise plus T_EX comme moteur. `pdfetex` est un moteur capable de produire *exactement la même sortie* que T_EX si nécessaire, mais dispose d'extensions (contenues soit dans `etex` soit dans `pdftex`) permettant, par exemple l'écriture de droite à gauche, l'alignement optique des marges, de produire une sortie `.pdf` au lieu d'un `.dvi`, etc. Le format `latex.fmt`, produit toujours un fichier `.dvi`.

11. Ceci permet d'avoir des caractères lisibles dans les fichiers `.log` à la place des codes hexadécimaux `^xy`.

Fontes METAFONT: le fichier `mktex.cnf`

Lorsque des polices au format METAFONT sont utilisées dans un document ¹², il est nécessaire de les convertir au format « bitmap » pour les imprimer ou les visualiser, aucun pilote de sortie n'étant capable de les traiter directement.

Le fichier `mktex.cnf` contient les définitions des paramètres nécessaires à METAFONT pour fabriquer les polices « bitmap » à partir des fichiers `.mf`.

Les deux paramètres essentiels sont le type de l'imprimante par défaut (variable `MODE`) et sa résolution (variable `BDPI`). Les valeurs possibles figurent dans le fichier `texmf/metafont/misc/modes.mf`, le script `texconfig` les affiche directement à l'écran pour un choix plus facile.

Les pilotes de visualisation sur écran (`xdvi` par exemple) utilisent les polices « bitmap » créées pour l'imprimante (ou une des imprimantes) en les réduisant : voir le fichier `texmf/xdvi/XDvi`. Si on change d'imprimante et qu'on passe par exemple d'une résolution de 600dpi (la valeur par défaut actuellement) à 1200dpi, il vaut mieux supprimer toutes les fontes « bitmap » créées pour l'ancienne imprimante, et modifier le fichier `XDvi` de façon à lui faire prendre en compte les mêmes polices « bitmap » que la nouvelle imprimante (paramètres `*mfMode` et `*shrinkFactor`).

Fontes vectorielles : le fichier `updmap.cfg`

Le fichier `updmap.cfg` contient la liste des fichiers `.map` à faire prendre en compte par `dvips`, `pdftex` et `dvipdfm`. Chacun de ces fichiers `.map` contient une liste de fontes, type 1 ou TrueType, qui doivent être utilisées directement sans création de polices « bitmap ».

Le fichier `updmap.cfg` est à modifier à chaque fois qu'on installe de nouvelles fontes type 1 ou TrueType. Ceci peut être fait, comme on l'a vu à la section 4 pour l'installation de Fourier, grâce à la commande

```
updmap-sys --enable Map fourier.map
```

qui ajoute la ligne « `Map fourier.map` » au fichier `updmap.cfg` « actif » dont on trouve le chemin complet en lançant la commande

```
kpsewhich --format='web2c files' updmap.cfg
```

12. La majorité des polices utilisées avec \TeX existant maintenant en format type 1, il faut éviter autant que possible le recours aux polices METAFONT qui, rappelons-le, s'affichent mal à l'écran dans les fichiers `.pdf`.

Ce fichier se trouve par défaut dans l'arborescence principale¹³, les ajouts qui y sont faits de cette manière seront à refaire à chaque changement de version de TeXLive.

Personnellement, je préfère regrouper tous les fichiers modifiant la configuration standard dans le répertoire TEXMFLOCAL. Voici comment je procède pour ne pas avoir à compléter le fichier `updmap.cfg` à chaque changement de version de TeXLive: je crée sous TEXMFLOCAL/web2c un fichier `updmap.add` qui contient les lignes qui manquent dans `updmap.cfg` pour prendre en compte les fontes locales, type 1 ou TrueType, par exemple :

```
Map utopia.map
Map fourier-utopia-expert.map
```

et je me suis fait un petit script `updmap-local` qui fusionne le fichier `updmap.cfg` de la distribution TeXLive et le fichier local `updmap.add` puis lance `updmap-sys` avec le fichier résultant comme fichier de configuration. Voici ce que contient le script `updmap-local` :

```
#!/bin/sh
cfg='kpsewhich --format='web2c files' updmap.cfg'
add='kpsewhich --format='web2c files' updmap.add'
tmp=/tmp/updmap.merge
if [ -f $tmp ]; then
    \rm -f $tmp
fi
cat $cfg > $tmp
cat $add >> $tmp
updmap-sys --cnffile $tmp
```

Cette solution ne nécessite pas de définir `TEXMFSYSCONFIG` comme `TEXMFLOCAL`. Une autre solution, avec les valeurs de `TEXMFSYSCONFIG` et `TEXMF` données page 15, consiste à placer une version complétée de `updmap.cfg` dans `TEXMFLOCAL/web2c` et à lancer `updmap-sys`. Ceci ne dispense toutefois pas de mettre à jour chaque année la partie standard du fichier `updmap.cfg` local.

Réglages divers : retour sur le fichier `texmf.cnf`

Dans la section 1, nous nous sommes limités à la présentation de la première partie du fichier `texmf.cnf`, celle qui définit les chemins de recherche des fichiers.

La deuxième partie définit les options non liées aux chemins, nous allons en passer en revue quelques unes :

13. Dans la distribution TeXLive 2008, il est sous `TEXMFSYSVAR`, alors qu'il devrait être sous `TEXMFSYSCONFIG`.

`shell_escape = f` ; par défaut il n'est pas possible de lancer des commandes *shell* par l'intermédiaire de `\write18{...}`, ceci pour des raisons de sécurité (la compilation sous Unix d'un fichier \LaTeX contenant la ligne `\write18{rm -R $HOME}` aurait pour effet d'effacer tout le répertoire personnel de l'utilisateur!). Cependant il peut être nécessaire d'autoriser l'exécution de commandes *shell*, c'est par exemple indispensable pour utiliser l'extension *emp* qui permet d'inclure le code `METAPOST` des figures dans un source \LaTeX (la commande `mpost` est lancée par un `\write18`). On peut le faire soit globalement en remplaçant dans `texmf.cnf` la valeur « f » de `shell_escape` par « t » (pas vraiment dangereux sauf si on compile n'importe quel fichier sans en vérifier le contenu), soit au coup par coup en lançant la commande « `latex -shell_escape` » au lieu de « `latex` ».

`openout_any = p` ; ceci interdit à \TeX d'écrire dans deux types de fichiers :

- a) ceux dont le nom commence par « . » (ce sont les fichiers de configuration sous Unix comme `.bashrc`, `.emacs`, etc.),
- b) ceux qui sont situés en dehors du répertoire courant ou de ses sous-répertoires¹⁴.

Ce choix est justifié par des raisons de sécurité. On peut donner à `openout_any` la valeur « r » pour lever la restriction b), ou « a » pour autoriser l'écriture dans tous les fichiers (franchement déconseillé!).

`openin_any = a` ; la lecture de tous les fichiers, où qu'ils se trouvent, est autorisée par défaut. Il est possible de restreindre les autorisations en lecture avec les valeurs « r » ou « p » comme ci-dessus, mais rien ne le justifie à mon avis.

`TEXEDIT =` ; cette variable, non définie par défaut, est utile lorsqu'on lance les compilations à partir d'un terminal. Elle permet d'indiquer quel éditeur utiliser lorsqu'on tape « e » pour éditer le texte source à la suite d'une erreur : le curseur vient se placer dans la ligne provoquant l'erreur. De nombreux exemples sont donnés dans le fichier `texmf.cnf` pour divers éditeurs sous Windows et sous Linux.

La troisième partie du fichier `texmf.cnf` définit la taille des tableaux internes utilisés par \TeX . Les tailles choisies pour TeX Live 2008 sont largement suffisantes en pratique, il n'y a normalement pas lieu de les modifier.

14. L'écriture reste toutefois autorisée dans les fichiers qui se trouvent dans l'arborescence désignée par la variable `TEXMFOUTPUT` (également définie dans `texmf.cnf` par défaut c'est `/tmp`).

8 Cohabitation de plusieurs distributions TeX sous Unix

Nous avons évoqué à la section 3, le cas de l'installation de TeXLive 2008 en plus de la version 2007 installée de base sous Debian Sarge.

Compte tenu du faible coût actuel de l'espace disque¹⁵, il peut s'avérer utile de conserver plusieurs distributions TeX sur une même machine, par exemple les distributions TeXLive de l'année en cours et celle d'une ou plusieurs années précédentes. Ceci permet de recomposer à l'identique des anciens documents mais surtout de passer en douceur d'une distribution à la suivante. L'administrateur d'une machine peut installer une nouvelle version de TeXLive, la tester à sa guise seul ou avec quelques collègues, et basculer l'ensemble des utilisateurs sur la nouvelle version le moment venu. Il est aussi possible à tout moment pour un utilisateur particulier (ou pour tous) de revenir à une version précédente. Les mises à jour se font ainsi sans aucun risque.

Nous ne reviendrons pas sur l'installation de TeXLive qui est largement décrite dans le fichier `live.pdf` joint à la distribution. Depuis TeXLive 2008, l'installation et les mises à jour se font très facilement par la réseau.

La *seule* chose à modifier pour passer d'une distribution TeX à une autre est le contenu de la variable d'environnement PATH et éventuellement celui de TEXMFCNF si elle a été définie (voir page 15).

La syntaxe dépend de l'interpréteur de commandes (*shell* en anglais) utilisé (en cas de doute, la commande `echo $SHELL` retourne son nom) :

— si c'est bash ou sh,

```
export PATH=<chemin-complet-des-binaires>:$PATH
```

— si c'est tcsh ou csh,

```
setenv PATH <chemin-complet-des-binaires>:$PATH
```

Pour TeXLive 2008, le chemin complet des binaires Linux est par défaut : `/usr/local/texlive/2008/bin/i386-linux/`.

Lorsqu'on exécute une de ces commandes dans un terminal, le changement de distribution TeX n'est effectif que pour la session en cours. Si on veut retrouver les mêmes binaires TeX automatiquement lors des sessions suivantes, il faut mettre ces commandes dans un fichier lu à la connexion. Ça peut être `.bash_profile` (*shell* bash) ou `.profile` (*shell* sh) ou `.tcshrc` (*shell* tcsh) ou `.cshrc` (*shell* csh)... ces fichiers se trouvent à la racine du répertoire personnel de chaque utilisateur.

Les administrateurs de serveurs mettent en général au début des fichiers lus à la connexion (`.bash_profile` et consorts) une ligne du type « `source fichier` » qui leur permet d'initialiser les principales variables d'environnement (dont PATH)

15. Une distribution comme TeXLive 2008 en version GUTenberg occupe environ 1 Go...

sans toucher aux fichiers personnels des utilisateurs. Il leur suffit alors de modifier le *fichier* de référence pour faire passer tous les utilisateurs d'une distribution \TeX à un autre, sans devoir intervenir sur chaque compte. Ceci n'empêche pas l'utilisateur de modifier la variable `PATH` à sa guise, il lui suffit de le faire après la ligne « source *fichier* ».

Remarque importante : il est maintenant possible de *remplacer* la distribution \TeX officielle incluse dans Debian, par une distribution plus récente, TeXLive 2008 par exemple. Ceci évite les confusions et problèmes dus à la présence de plusieurs distributions \TeX . La façon de procéder est décrite sur le blog de Manuel Pégourié-Gonnard : <http://weblog.elzevir.fr/2008/11/tex-live-2008-a-la-vanille-sur-debian-like/>.

9 TeX Live intégrée aux distributions Debian

Installation de TeX Live en .deb

L'installation est particulièrement simple : la commande (à exécuter sous *root*)
`aptitude install texlive lmodern latex-beamer latex-ucs`
fournit une installation standard. On peut remplacer `texlive` par `texlive-full` pour obtenir une installation exhaustive (solution de facilité pour ceux qui ont pléthore d'espace disque), ou bien compléter, si nécessaire, l'installation de base en ajoutant les seuls paquets jugés utiles. On trouvera la liste de tous les paquets composant `texlive-full` en tapant la commande
`aptitude search texlive`

Après l'installation, vérifier que la commande « `which pdftex` », exécutée par un utilisateur autre que *root*, retourne `/usr/bin/pdftex`, si ce n'est pas le cas il convient de modifier le contenu de la variable `PATH` : une autre distribution TeX (ancienne TeX Live ?) doit cacher la distribution en .deb.

Transfert d'une arborescence locale préexistante

Ceux qui ont déjà une installation fonctionnelle de TeXLive avec une arborescence locale devront déplacer (ou recopier) celle-ci pour qu'elle soit prise en compte par la distribution TeXLive en .deb. Cette étape est bien sûr inutile si rien n'a été installé sous l'ancienne arborescence locale.

Pour localiser la nouvelle arborescence locale, taper la commande
`kpsewhich -var-value=TEXMFLOCAL`
elle retourne normalement `/usr/local/share/texmf`. Pour localiser l'ancienne arborescence locale, lorsqu'on a oublié où elle trouve, c'est un peu plus

compliqué ; on peut procéder ainsi : retrouver d'abord le répertoire des binaires de l'ancienne distribution TeXLive en tapant la commande

```
locate kpsewhich | grep bin
```

celle-ci retournera au moins deux lignes, celle contenant `/usr/bin/kpsewhich` correspond à l'exécutable `kpsewhich` de la distribution TeXLive en `.deb`. L'ancienne distribution devait normalement être installée sous `/usr/local` ou sous `/opt`, ou ... ce qui doit permettre de conclure. Supposons que l'exécutable `kpsewhich` de l'ancienne distribution T_EX était sous, disons, `/usr/local/texlive/2005/bin/i386-linux/`, on tape alors la commande `/usr/local/texlive/2005/bin/i386-linux/kpsewhich -var-value=TEXMFLOCAL` pour obtenir le chemin de l'ancienne arborescence locale. Mettons que le résultat soit `/usr/local/texlive/2005/./texmf-local`, c'est-à-dire le répertoire `/usr/local/texlive/texmf-local` (le symbole `..` fait remonter d'un niveau dans la hiérarchie). Il faut dans ce cas déplacer ou recopier chacun des répertoires contenus dans `/usr/local/texlive/texmf-local` dans la nouvelle arborescence locale (`/usr/local/share/texmf` dans notre exemple).

Il convient de supprimer de TEXMFLOCAL les composants qui ont été intégrés entre-temps à la nouvelle distribution T_EX et ceux qui y figurent dans une version plus récente (il arrive que des vieilleries traînent dans TEXMFLOCAL, le changement de distribution est une bonne occasion de faire le ménage!).

Enfin, il faut mettre à jour la base `ls-R` de TEXMFLOCAL en tapant la commande `mktexlsr` et éventuellement faire prendre en compte les fontes type 1 ou `truetype` contenues dans TEXMFLOCAL grâce à la commande `updmap-sys` (voir la section « Personnalisation de l'installation » ci-dessous).

Le chemin des répertoires `texmf` personnels des utilisateurs ne devrait pas avoir changé, il se trouve normalement sous `$HOME/texmf` (à vérifier toutefois).

Personnalisation de l'installation

Debian recommande de *ne surtout pas éditer les fichiers de configuration* `texmf.cnf`, `fmtutil.cnf`, `updmap.cfg`, etc. Des procédures spécifiques ont été mises en place pour éviter que les modifications faites localement ne soient écrasées lors des mises à jour de la distribution TeXLive. Nous allons décrire maintenant ces procédures.

On trouve sous `/etc/texmf` quatre répertoires dont le nom se termine par `.d`

- `fmt.d`; le fichier `fmtutil.cnf` est obtenu par concaténation des fichiers présents dans `/etc/texmf/fmt.d` lors de l'exécution de la commande `updatefmtutil`.

- `language.d`; le fichier `language.dat` est obtenu par concaténation des fichiers présents dans `/etc/texmf/language.d` lors de l'exécution de la com-

mande `update-language`.

— `texmf.d`; le fichier `texmf.cnf` est obtenu par concaténation des fichiers présents dans `/etc/texmf/texmf.d` lors de l'exécution de la commande `update-texmf`.

— `updmap.d`; le fichier `updmap.cfg` est obtenu par concaténation des fichiers présents dans `/etc/texmf/updmap.d` lors de l'exécution de la commande `update-updmap`.

Pour modifier l'un des fichiers `fmtutil.cnf`, `language.dat`, `texmf.cnf`, `updmap.cfg`, il convient donc d'ajouter des fichiers dans l'un des quatre répertoires de la liste ci-dessus et de lancer la commande de régénération correspondante. Exemples :

Prise en compte de fontes type 1 locales — On suppose que les fichiers `.tfm`, `.vf`, `.pfb`, `.fd`, `.sty` et `.map`, ont été préalablement correctement installés comme indiqué p. 12 pour les fontes Fourier et que la base de données a été mise à jour (commande `mktexlsr`). Il reste à mettre à jour le fichier `updmap.cfg`, ce qui se fait, sous Debian, en ajoutant un fichier, par exemple `90localmap.cfg`¹⁶ qui contiendra les noms des fichiers `.map` à prendre en compte, par exemple

```
Map utopia.map # fontes utopia Adobe
Map ugm.map    # fontes Garamond URW
Map mdugm.map # fontes MathDesign pour Garamond URW
```

On termine en lançant la commande `update-updmap` qui régénère le fichier `updmap.cfg`, puis `update-sys` qui régénère les fichiers `psfonts.map` (pour `dvips`), `pdftex.map` (pour `pdftex`) et `dvipdfm.map` (pour `dvipdfm`).

Modification du fichier `texmf.cnf` — Supposons qu'on veuille modifier la valeur de certaines variables définies dans `texmf.cnf`, par exemple

```
TEXEDIT = emacsclient --no-wait +%d %s
shell_escape = t
```

On crée un fichier `00localvar.cnf`¹⁷ contenant ces deux lignes, on exécute ensuite la commande `update-texmf` qui régénère `texmf.cnf` et c'est tout car ce fichier est lu à chaque exécution d'une commande de la famille TeX, latex, dvips, kpsewhich, etc.

Ces procédures peuvent paraître lourdes mais elles ont l'immense avantage d'assurer la pérennité des ajouts et modifications locales sans aucune intervention de l'administrateur lors des mises à jour système faite par aptitude upgrade.

16. Les noms des fichiers de `/etc/texmf/updmap.d` commencent par deux chiffres qui définissent l'ordre de concaténation. Le préfixe 90 du nom `90local.cfg` placera les fontes locales à la fin du fichier `updmap.cfg`.

17. Il est primordial que ce fichier soit pris en compte *avant* les autres lors de la création de `texmf.cnf`, car la valeur d'une variable déjà positionnée dans `texmf.cnf` n'est jamais modifiée par la suite, d'où le préfixe 00. Un fichier `90localvar.cnf` n'aurait *aucun effet*.

Références

- [1] La TDS en français, cahiers GUTenberg n° 44-45
<http://www.gutenberg.eu.org/publications/cahiers/>
- [2] La TDS en v. o., <http://tug.org/tds/tds.pdf>
- [3] Installation de TeXLive, texmf-doc/doc/french/texlive-fr/live.pdf
ou sur le DVD : texlive/texmf-doc/doc/french/texlive-fr/live.pdf
- [4] TeXLive en .deb pour Debian : consulter (sous un navigateur) le fichier
[/usr/share/doc/tex-common/TeX-on-Debian.html](http://usr/share/doc/tex-common/TeX-on-Debian.html) ou la version PDF
compressée [TeX-on-Debian.pdf.gz](http://usr/share/doc/tex-common/TeX-on-Debian.pdf.gz) dans le même répertoire.
- [5] Bernard Desgraupes, *LaTeX apprentissage, guide et référence* Vuibert.

N'hésitez pas à me faire part de vos
commentaires et suggestions sur
ce texte qui a vocation à être
complété et amélioré.



Daniel Flipo,
juillet 2009.