

# *Cahiers* **GUT** *enberg*

☞ RAPIDITÉ ET SOUPLESSE AVEC LE MOTEUR  
WEB2C-7

☞ Fabrice POPINEAU

*Cahiers GUTenberg*, n° 26 (1997), p. 96-108.

<[http://cahiers.gutenberg.eu.org/fitem?id=CG\\_1997\\_\\_26\\_96\\_0](http://cahiers.gutenberg.eu.org/fitem?id=CG_1997__26_96_0)>

© Association GUTenberg, 1997, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*

(<http://cahiers.gutenberg.eu.org/>),

implique l'accord avec les conditions générales

d'utilisation (<http://cahiers.gutenberg.eu.org/legal.html>).

Toute utilisation commerciale ou impression systématique

est constitutive d'une infraction pénale. Toute copie ou impression

de ce fichier doit contenir la présente mention de copyright.

---

# Rapidité et souplesse avec le moteur Web2c 7

---

Fabrice POPINAU

*SUPELEC, Campus de Metz*

*email: popineau@esemetz.esemetz.fr*

## 1. Qu'est-ce que Web2c?

C'est une question fréquente dans les milieux où l'on parle de  $\TeX$ . A l'origine, D.E. Knuth a écrit  $\TeX$  dans un langage de programmation plutôt inhabituel nommé *Web*. L'originalité de ce langage est de permettre la programmation mot-à-mot (*Literate programming*): les algorithmes sont décrits in extenso, puis la documentation et le code sont extraits par deux outils séparés. A l'origine, le code *Web* était converti en *Pascal*, puis compilé. A l'heure où *Pascal* n'est plus tellement en vogue, et le langage *C* universellement répandu, un traducteur de *Web* vers *C* a naturellement été développé.

Historiquement, Tomas Rokicki<sup>1</sup> est à l'origine du premier système  $\TeX$ -to-*C* en 1987. Tim Morgan a continué de maintenir ce système pendant quelques années, durant lesquelles le nom a changé pour *Web-to-C*. En 1990, Karl Berry a pris officiellement le relais, aidé par de nombreux bénévoles dont les interventions sont répertoriées dans le fichier `ChangeLog` de la distribution.

La dernière version de la distribution *Web2c* porte le numéro 7.0 et a été officiellement annoncée le 8 février 1997. La version précédente, numérotée 6.1 datait de janvier 1995. La version 7.0, apportant de nombreuses fonctionnalités nouvelles, souffre de défauts de jeunesse qui seront corrigés dans une très prochaine version 7.1.

La distribution *Web2c* version 7.0 complète se compose des parties suivantes :

- `kpathsea 3.0` : bibliothèque permettant la recherche des fichiers de style, de polices, ...

---

1. auteur du célèbre `dvips`

- convertisseur *Web2c*: génère du code *C* à partir du code *Web*. Le code généré n'est pas autosuffisant, mais repose sur une bibliothèque d'entrées/sorties particulière qui fait elle-même appel à *kpathsea* ;
- $\TeX$  3.14159, METAFONT 2.718, METAPOST 0.632: les programmes de base de la distribution ;
- Bibtex 0.99c ;
- les programmes utilitaires suivants: *dvicopy*, *dvitomp*, *dvitype*, *gftodvi*, *gftopk*, *gftype*, *mft*, *patgen*, *pktogf*, *pktype*, *pltotf*, *pooltype*, *tangle*, *tftopl*, *vftovp*, *vptovf*, *weave* ;
- *dvips* 5.66: le pilote d'impression Postscript ;
- *dvilj* 2.6: le pilote d'impression pour imprimantes LaserJet HP ;
- *xdvi* 20: le visualiseur de fichiers *dvi* pour X-Window.

L'ensemble de ces programmes compose un environnement  $\TeX$  complet pour Unix. A l'origine, chacun de ces programmes était paramétré individuellement par des variables d'environnement, dont la plupart spécifiaient des chemins de recherche pour tel ou tel type de fichiers. Les procédures de recherche étaient dupliquées dans des programmes maintenus par des auteurs différents et des variables d'environnement de même sémantique pouvaient porter des noms différents. Tout l'intérêt du travail de Karl Berry et de *Web2c* réside dans l'homogénéisation de l'interface des divers programmes avec le système et de leur paramétrage.

Il n'est pas question ici de remplacer les manuels fournis dans la distribution, mais seulement de donner quelques éléments pour apprécier les possibilités de *Web2c*. Nous parlerons donc successivement de *kpathsea* et de son paramétrage, des programmes de la distribution et des portages existants.

## 2. Bibliothèque *kpathsea*

### 2.1. Vous avez dit *kpathsea*?

Cette bibliothèque a été conçue pour rechercher un fichier dans une liste de répertoires spécifiés par l'utilisateur. Dans le cas de  $\TeX$ , l'arborescence pouvant être conséquente – plusieurs dizaines de milliers de fichiers répartis dans plusieurs milliers de répertoires – il est vital que cette tâche soit effectuée de la manière la plus efficace possible.

Karl Berry est l'auteur de cette bibliothèque et la maintient actuellement. Elle est utilisée par les programmes de `Web2c`, mais aussi par les `GNU font utilities` et par quelques autres logiciels.

Tous les programmes de la distribution `Web2c` utilisent `kpathsea` pour s'interfacer avec le système. L'homogénéité de la distribution réside dans cette bibliothèque. On peut envisager `kpathsea` sous plusieurs angles. D'une part, elle peut être considérée comme une base de données de fichiers `TEX` et de polices `TEX`, réalisant les opérations classiques permises par une base de données : recherche et insertion d'éléments<sup>2</sup>. D'autre part, on peut la considérer comme un modèle unifié et englobant de toutes les procédures de paramétrage et de toutes les interfaces système des programmes de `Web2c`. Un point crucial dans la conception de `kpathsea` a été – comme souvent en informatique – de permettre la mise à niveau en douceur à partir des anciennes versions de `TEX`. Par exemple, `tex` était configuré à l'aide de variables d'environnement telles que `$TEXINPUTS`, et c'est toujours le cas dans `Web2c`. Si la variable d'environnement est définie, elle sera prise en compte en premier lieu, et le fonctionnement de `tex` sera compatible avec l'ancienne version. Dans une installation plus récente, on fera disparaître de l'environnement des utilisateurs, les variables nécessitées par `TEX`, pour se reposer uniquement sur les paramètres définis dans le fichier `texmf.cnf`. Finalement, chaque utilisateur n'aura plus qu'une ou deux variables spécifiques à `TEX` au maximum dans son environnement, contre éventuellement plusieurs dizaines auparavant<sup>3</sup>.

La structure de l'arborescence des fichiers support de `TEX` ayant pu évoluer dans des directions très différentes selon les sites, l'intérêt de `kpathsea` repose également sur une homogénéisation de cette structure.

## 2.2. Arborescence standard de répertoires

Le TUG<sup>4</sup> a entériné une norme pour l'arborescence des fichiers utilisés par un système `TEX`. Cette norme s'appelle TDS pour `TEX Directory Structure`. La bibliothèque `kpathsea` est livrée préconfigurée pour chercher les fichiers dans une arborescence conforme à la TDS.

On peut accélérer notablement la recherche d'un fichier de manière importante en tenant compte du point suivant. Un fichier contient une information caractérisée par plusieurs critères. Ces critères sont classés et une arborescence consituée en utilisant les critères dans l'ordre choisi. Tous les fichiers conte-

---

2. La destruction de fichiers n'est pas intéressante dans ce contexte. L'insertion se produit lors de la création automatique de nouveaux fichiers de polices.

3. Voir ce qui peut être nécessité par `emTEX` par exemple.

4. *T<sub>E</sub>X User Group*

nant le même type d'information sont donc enregistrés dans le même répertoire. La recherche d'un fichier peut donc s'effectuer en suivant un chemin dans l'arborescence défini par les critères de ce fichier. C'est la théorie, car en pratique, il existe souvent plusieurs classements intéressants pour les critères. Par exemple, on peut choisir pour un fichier de police, de privilégier une famille (cm, ec, lucida, ...) par rapport au type de fichier (métrique, bitmap, ...) ou l'inverse.

La TDS correspond au résultat des réflexions d'un groupe de travail du TUG sur ce sujet. La TDS est une proposition de structure de répertoires pour un environnement  $\TeX$ , mais la bibliothèque *kpathsea* n'est aucunement liée à cette structure, et aucune implémentation de  $\TeX$  ne devrait l'être<sup>5</sup>.

Voici un exemple de TDS pour un système  $\TeX$  générique, exemple extrait du document définissant la TDS :

<code>bibtex/</code>	Bi $\TeX$ input files
<code>bib/</code>	Bi $\TeX$ databases
<code>base/</code>	base distribution (e.g., <code>xampl.bib</code> )
<code>misc/</code>	single-file databases
<code>&lt;package&gt;/</code>	name of a package
<code>bst/</code>	Bi $\TeX$ style files
<code>base/</code>	base distribution (e.g., <code>plain.bst</code> , <code>acm.bst</code> )
<code>misc/</code>	single-file styles
<code>&lt;package&gt;/</code>	name of a package
<code>doc/</code>	documentation files
<code>fonts/</code>	font-related files
<code>&lt;type&gt;/</code>	file type (e.g., <code>pk</code> )
<code>&lt;mode&gt;/</code>	type of output device (for <code>pk</code> and <code>gf</code> only)
<code>&lt;supplier&gt;/</code>	name of a font supplier (e.g., <code>public</code> )
<code>&lt;typeface&gt;/</code>	name of a typeface (e.g., <code>cm</code> )
<code>dpi&lt;nnn&gt;/</code>	font resolution (for <code>pk</code> and <code>gf</code> only)
<code>&lt;implementation&gt;/</code>	$\TeX$ implementations, by name (e.g., <code>emtex</code> )
<code>metafont/</code>	METAFONT (non-font) input files
<code>base/</code>	base distribution (e.g., <code>plain.mf</code> )
<code>misc/</code>	single-file packages (e.g., <code>modes.mf</code> )
<code>&lt;package&gt;/</code>	name of a package (e.g., <code>mfpic</code> )
<code>metapost/</code>	METAPOST input and support files
<code>base/</code>	base distribution (e.g., <code>plain.mp</code> )
<code>misc/</code>	single-file packages
<code>&lt;package&gt;/</code>	name of a package
<code>support/</code>	support files for METAPOST-related utilities
<code>mft/</code>	MFT inputs (e.g., <code>plain.mft</code> )

5. D'ailleurs, la TDS résulte d'un compromis dont certains points sont discutables et sur lesquels Karl Berry lui-même n'était pas d'accord.

---

<code>&lt;program&gt;/</code>	TeX-related programs, by name (e.g., <code>dvips</code> )
<code>source/</code>	program source code by name (e.g., <code>latex</code> , <code>web2c</code> )
<code>tex/</code>	TeX input files
<code>&lt;format&gt;/</code>	name of a format (e.g., <code>plain</code> )
<code>base/</code>	base distribution for format (e.g., <code>plain.tex</code> )
<code>misc/</code>	single-file packages (e.g., <code>webmac.tex</code> )
<code>local/</code>	local additions to or local configuration files for <code>&lt;format&gt;</code>
<code>&lt;package&gt;/</code>	name of a package (e.g., <code>graphics</code> , <code>mfnfss</code> )
<code>generic/</code>	format-independent packages
<code>hyphen/</code>	hyphenation patterns (e.g., <code>hyphen.tex</code> )
<code>images/</code>	image input files (e.g., Encapsulated PostScript)
<code>misc/</code>	single-file format-independent packages (e.g., <code>null.tex</code> ).
<code>&lt;package&gt;/</code>	name of a package (e.g., <code>babel</code> )

Parallèlement à la TDS, un système de normalisation des noms de polices a été développé sur l'impulsion de Karl Berry. Ce système prévoit que tout fichier de police porte un nom qui rappelle la famille, le type, la taille, etc... Et ceci sans dépasser 8 caractères<sup>6</sup>!

On est en droit de se demander pourquoi développer un système aussi complexe pour gérer entre autres les fichiers de polices. Pour simplifier, pourquoi ne pas mettre tous les `tfm` dans un répertoire et tous les `pk` dans un autre? Cette question possède deux réponses :

1. il est plus satisfaisant pour des raisons de maintenance, de classer les fichiers de polices. La mise à jour s'en trouve facilitée. Il peut y avoir des conflits de noms sur certains fichiers : par exemple, le fichier `cmr10.600pk` n'indique pas pour quelle imprimante il a été généré, en dehors du fait qu'il s'agit d'un moteur 600dpi.
2. plus important, l'installation de polices Postscript peut provoquer un accroissement très important du nombre de fichiers de polices – jusqu'à plusieurs dizaines de milliers. A ce stade, les performances de recherche à l'intérieur d'un répertoire contenant 10000 fichiers deviennent très mauvaises. Une arborescence plus profonde contenant moins de fichiers par répertoire est beaucoup plus efficace. Le manuel de `kpathsea` décrit certaines astuces<sup>7</sup> mises en œuvre sous Unix entre autres pour accélérer encore la recherche, par exemple en vérifiant que les nœuds intermédiaires de l'arborescence contiennent seulement des répertoires et pas de fichiers.

---

6. A cause des CD-Rom et de MS-DOS.

7. Pour avoir les explications les plus pointues, il faut bien sûr se reporter au code ...

Les fichiers décrivant le système de nommage de fichiers de police et permettant de l'exploiter sont situés dans `texmf/fontname`. Le système `fontname`, puisque tel est son nom, est d'une aide précieuse pour archiver les fichiers de polices : couplé à la TDS, le nom seul du fichier permet d'accélérer encore la recherche en fournissant des indices sur sa localisation. D'autre part, lors de la génération de nouveaux fichiers de polices, on utilise également le nom du fichier généré pour déterminer sa localisation future.

### 2.3. Possibilités de configuration

La configuration du système Web2c est centralisée autour d'un nombre restreint de fichiers : `texmf.cnf`, `MakeTeX.site` et `MakeTeXnames.cnf`. Le premier configure la recherche des fichiers dans l'arborescence `texmf`, les deux autres sont utiles pour la génération de fichiers de polices. Cette centralisation des variables de configuration a été rendue partiquement obligatoire par leur prolifération dans des logiciels maintenus par des personnes différentes, variable qui pouvaient même porter des noms différents<sup>8</sup>.

#### 2.3.1. Fichier de configuration `texmf.cnf`

Le fichier fourni par défaut rassemble la quasi-totalité des nombreuses possibilités. Ce fichier contient un grand nombre de couples `variables = valeur`, qui dans d'autres systèmes, correspondent à autant de variables d'environnement. Tout programme de Web2c – reposant sur `kpathsea` – n'a besoin que de pouvoir trouver ce fichier. Si la distribution a été compilée sur place, les paramètres fournis à la compilation suffisent. En principe, ce fichier devrait pouvoir être trouvé automatiquement en testant diverses positions relatives à la position du fichier binaire demandeur. Si l'arborescence adoptée est suffisamment standard, le fichier `texmf.cnf` sera localisé automatiquement. Sinon, il suffit de faire pointer la variable `$TEXMFCNF` sur le fichier `texmf.cnf`.

Chacune des variables de configuration possède une valeur par défaut définie au moment de la compilation. Si cette variable n'est pas définie dans `texmf.cnf`, la valeur à la compilation est prise par défaut. En revanche, si une variable d'environnement du même nom est définie, alors cette variable d'environnement masque la valeur du fichier de configuration.

Ces variables de configurations sont de deux types : ce sont soit des chemins pour accéder à certains types de fichiers, soit des paramètres liés à l'exécution de programmes.

---

8. Ce qui a été le cas pour `tex` et `dvips` où des variables de noms différents avaient la même fonction

### 2.3.2. Recherche sur les chemins

La bibliothèque `kpathsea` peut faire usage d'une base de données pour accélérer la recherche de fichiers. Si un fichier de nom `ls-R` est présent à la racine de l'arborescence `texmf`, et que ce fichier `ls-R` possède le bon format, alors son contenu est lu et mémorisé dans une table à accès direct. Ce fichier `ls-R` n'est que le résultat de la commande `'ls -R'` sous Unix. Par la suite, la recherche d'un fichier s'effectuera d'abord dans la table à accès direct, avant une recherche sur disque.

Les chemins spécifiés pour rechercher les fichiers peuvent faire usage de caractères génériques. Analysons l'exemple suivant :

```

TEXMFDBS = /usr/local/texmf

% The TeX inputs and fonts directories.
texdir = $TEXMF/tex
fontdir = $TEXMF/fonts

% To avoid searching the filesystem,
%       use the commented-out definitions here.
%dbtex = !!$texdir
%dbfonts = !!$fontdir
dbtex = $texdir
dbfonts = $fontdir

% TeX input files -- i.e., anything to be found by \input
%                                     or \openin,
% including .sty, .eps, etc.
TEXINPUTS = . ;$dbtex//

VFFONTS = . ;$dbfonts//vf//

```

Par le jeu de substitution des variables, `VFFONTS` vaut `/usr/local/texmf/fonts//vf//`. Il faut interpréter ce chemin comme un motif dont la signification est : « chercher les fichiers de polices virtuelles sur tous les chemins qui commencent par `/usr/local/texmf/fonts` et qui contiennent `/vf/` dans la suite ».

Chaque fichier est cherché en premier lieu dans la base de données qui a été lue en mémoire centrale, puis si il est trouvé, son existence réelle est testée. Si

la base de données ne le référence pas, une recherche disque est éventuellement entreprise. Cette recherche disque peut être interdite chemin par chemin en spécifiant !! au début du chemin. Il est ainsi possible de n'effectuer que des recherches dans la base de données, pourvu que celle-ci soit à jour. L'efficacité du système est alors maximale.

### 2.3.3. *Arborescences multiples*

C'est une nouveauté introduite par la version 3.0 de *kpathsea* qui a permis d'envisager l'utilisation d'arborescences *texmf* multiples. Dorénavant, *kpathsea* permet d'utiliser une notation similaire à celle d'un interpréteur de commandes (*shell*) Unix : `{foo,bar}` est expansé dans un chemin en une alternative `foo :bar`. Si les accolades sont prises au milieu d'un chemin, les alternatives sont reconstruites, ainsi on peut définir :

```
TEXMF = /usr/local/{texmf_fr,texmf}
TEXINPUTS = $TEXMF/tex//
```

La variable `$TEXINPUTS` sera réellement expansée en `/usr/local/texmf/tex// : /usr/local/texmf_fr/tex//`. Il suffit donc de changer une seule variable (`TEXMF`) pour permettre l'utilisation simultanée de plusieurs arborescences.

Ceci facilitera l'utilisation de distributions sur CD-Rom par exemple : il suffira de mentionner une arborescence fixe sur le CD-Rom plus une arborescence utilisateur, dans laquelle pourront avoir lieu des générations de fichiers de polices par exemple.

On peut également penser à modifier la configuration sur la base d'une langue : admettons que `/usr/local/texmf_fr` soit une partie d'arborescence à la norme TDS qui contienne des fichiers nécessaires à la francisation (styles, formats  $\LaTeX$ , ...). Pour passer d'un système  $\TeX$  anglo-saxon à un système francisé, il suffit de passer de `TEXMF=/usr/local/texmf` à `TEXMF=/usr/local/{texmf_fr,texmf}`

### 2.3.4. *Configuration des programmes*

Le fichier `texmf.cnf` permet d'agir sur le comportement des programmes de la distribution de deux manières différentes.

Tout d'abord, il est possible pour chaque variable de configuration, de spécialiser cette variable en fonction du programme qui l'utilise. Chaque variable

définie par `texmf.cnf` peut se voir masquée par une variable du même nom portant comme extension, le nom du programme appelant. Par exemple :

```
% LaTeX 2e specific macros are stored in latex/, macros that can
% only be used with 2.09 in latex209/.
% In addition, we look in the directory latex2e,
% useful for macros that were written for 2.09 and do not
% mention 2e at all, but can be used with 2e.
latex209_inputs = . ;$dbtex/latex209// ;$dbtex/latex//
                                ;$dbtex/generic//
latex2e_inputs = . ;$dbtex/latex// ;$dbtex/latex209//
                                ;$dbtex/generic//
TEXINPUTS.latex209 = $latex209_inputs
TEXINPUTS.latex2e = $latex2e_inputs
TEXINPUTS.latex = $latex2e_inputs
```

Selon que le programme appelant sera `latex` ou `latex209`, la variable `$TEXINPUTS` considérée sera différente. La transition entre  $\LaTeX$  et  $\LaTeX_{2e}$  en est grandement facilitée !

Il est également possible de spécialiser une variable d'environnement pour un programme dans l'environnement d'exécution lui-même en exécutant par exemple :

```
setenv TEXINPUTS_latex2e /usr/local/texmf/tex//
```

La deuxième manière d'agir sur le comportement des programmes a trait à certains paramètres internes des programmes, tels que la taille des zones mémoire. Il n'est plus nécessaire d'avoir recours à une compilation de `tex` pour changer la taille de la zone mémoire allouée aux motifs de coupure, par exemple :

```
% US English, German, and Portuguese : 30000.
% German : 14000.
% US English : 10000.
%
trie_size = 64000
```

### 2.3.5. *Autres fichiers de configuration*

Nous avons jusqu'ici uniquement parlé de la recherche de fichiers dans une arborescence, or le système de polices utilisé par  $\TeX$  est tel que les fichiers

optimaux de polices sont générés par le pilote qui va imprimer le document. Comme une grande variété de polices, de tailles standard, de pilotes et de périphériques sont disponibles, il est impensable de générer une fois pour toutes tous ces fichiers. Etant donné que toutes les combinaisons ne sont pas utilisées, il est plus raisonnable de se reposer sur une génération au vol et à la demande des fichiers manquants.

La bibliothèque *kpathsea* fournit des procédures écrites en shell pour générer les fichiers de police *mf*, *tfm* et *pk*. Encore faut-il générer ces polices dans le bon mode d'imprimante et choisir leur destination correcte dans l'arborescence. Idéalement, en respectant la TDS, tout fichier possède une place dans l'arborescence. Les fichiers nouvellement générés devraient donc être installés à cette place. Ce n'est pas toujours possible pour diverses raisons : l'installation de base se trouve sur un CD-Rom, les utilisateurs n'ont pas accès en écriture pour le répertoire de destination ... Le rôle des fichiers *MakeTeXnames.cnf* et *MakeTeX.site* est de palier tous ces problèmes et de permettre à chacun la définition d'un schéma rationnel quelle que soit sa configuration de base.

## 3. Programmes

### 3.1. Points communs

Tous les programmes partagent une interface commune d'appel qui reconnaît un certain nombre d'options :

- `--help` imprime une aide à l'utilisation du programme ;
- `--verbose` affiche la progression du traitement ;
- `--version` imprime la version du programme.

Tous les programmes qui ont besoin de rechercher des fichiers sur disque le font à l'aide de *kpathsea*.

### 3.2. *TeX*, METAFONT et METAOT

Les trois principaux programmes de la distribution se partagent d'autres options :

- `-kpathsea-debug=number` affiche une trace des opérations entreprises par *kpathsea* en fonction des bits positionnés dans le nombre fourni en argument ;

- **ini** lance une forme *initiale* du programme, ce qui est également fait implicitement si le programme s'appelle `initex` (resp. `inimf` et `inimpost`);

- **fmt=dumpname**

- **base=dumpname**

- **mem=dumpname** utilise le fichier de format de nom *dumpname*

Les formes *initiales* et *vierges* de chacun de ces programmes ont été unifiées. La distinction se fait par le nom d'appel ou par la présence de l'option `-ini`. La forme *initiale* de ces programmes permet de créer des images mémoires ou *formats* qui sont ensuite réutilisés par la forme *vierge*. C'est ainsi que le moteur `tex` permet d'obtenir le « plain T<sub>E</sub>X » avec un format `plain.fmt` ou bien L<sup>A</sup>T<sub>E</sub>X avec un format `latex.fmt`.

Un point intéressant concernant les formats : ils sont partageables entre architectures. Une version de Web2c 7.0 installée en réseau ne nécessite de différencier que les binaires.

Mentionnons encore quelques autres particularités de ces programmes :

- les trois programmes peuvent expander les caractères `~` et `$` à la manière du shell dans les commandes `\input` ;
- la manière dont T<sub>E</sub>X construit le nom des ses fichiers de sortie est configurable : est-ce que `foo.bar.tex` doit produire `foo.dvi` ou `foo.bar.dvi` ?
- M<sup>L</sup>T<sub>E</sub>X est intégré à T<sub>E</sub>X et appellable à la génération d'un format par l'option `-mltex`, un format généré avec cette option est reconnu automatiquement à son utilisation ;
- T<sub>E</sub>X peut exécuter directement des commandes système par l'intermédiaire de la commande `\write18{shell-command}` ;
- bien que ce ne soit pas fourni en standard dans Web2c, e-T<sub>E</sub>X est intégrable directement dans la version courante de Web2c et englobe par exemple `TeX-XTeX` ;
- T<sub>E</sub>X et METAFONT ont accès via `kpathsea` à la création de polices manquantes ;

### 3.3. Pilotes

Le programme `dvips` est maintenu par Tomas Rockicki. La dernière version est constamment disponible sur `ftp://labrea.stanford.edu:pub`. La version fournie avec Web2c est adaptée pour utiliser `kpathsea` plutôt que des fonctions internes de recherche de fichiers.

Le pilote pour imprimantes HP semble beaucoup moins utilisé.

Le visualiseur pour X-Window `XDvi` a été beaucoup amélioré et permet de gérer des liens hypertextuels. En revanche, sa maintenance semble devenir de plus en plus délicate !

### 3.4. Autres programmes

Les nombreux autres programmes fournis sont d'une utilisation moins fréquente mais complètent une installation :

- `bibtex` permet d'automatiser le traitement des bibliographies et de réutiliser les entrées dans d'autres documents ;
- la programmation *mot-à-mot* est supportée par `tangle` (traduction de Web en Pascal), `weave` (traduction de Web en  $\TeX$ ) et `pooltype` (impression de fichiers `pool`) ;
- `MakeMPX`, `DVItoMP`, `DMP`, `MPto` et `newer` sont des utilitaires dont le système `METAPOST` se sert ;
- les programmes `DVIcopy` et `DVItype` manipulent des fichiers `dvi`
- `gftopk`, `pkto gf`, `pktype`, `gf type`, `tftopl`, `pltotf`, `vftovp`, `vptovf` sont des utilitaires de conversion et d'affichage pour les différents formats de fichiers de polices.

## 4. Portages

La disponibilité de Web2c sur une grande variété de plateformes est un intérêt majeur. Ainsi, on peut l'utiliser sur la quasi-totalité des Unix, mais également sur Amiga et sur Windows95/NT (ou Win32).

La prochaine version du CD-Rom  $\TeX$ -Live distribué par le TUG et GUTenberg devrait utiliser une arborescence unique contenant les binaires pour les machines Unix les plus courantes, pour Amiga et pour Win32, ainsi qu'une arborescence `texmf` très complète.

La distribution Unix de  $\TeX$  la plus courante est certainement  $\text{teTeX}$  par Thomas Esser. Cette distribution est basée sur  $\text{Web2c}$  et contient les fichiers de styles et les polices les plus courantes pour  $\TeX$ . Elle possède également un outil de configuration très puissant.

La version  $\text{AmiWeb2c}$  pour Amiga est maintenue par Andreas Scherer ([Scherer@Physik.RWTH-Aachen.De](mailto:Scherer@Physik.RWTH-Aachen.De)).

La version  $\text{Win32}$  est maintenue par moi-même ([Fabrice.Popineau@supelec.fr](mailto:Fabrice.Popineau@supelec.fr)).

## 5. Conclusion

En conclusion, le système  $\text{Web2c}$  est certainement l'une des plateformes  $\TeX$  les plus ouvertes. Le développement actif qui tourne autour de cette plateforme permet d'ores et déjà d'envisager l'avenir :

- avec des extensions vers PDF, puisque  $\text{PDF}\TeX$  se construit au dessus de  $\text{Web2c}$ ,
- avec des extensions vers les polices 16 bits et  $\text{Omega}$ , puisque  $\text{kpathsea}$  supporte déjà la recherche des nouveaux formats de fichiers nécessaires, et que la version courante de  $\text{Omega}$  se construit également au-dessus de  $\text{Web2c}$ .

Le succès de  $\text{Web2c}$  est dû en grande partie à sa souplesse d'adaptation à de nouvelles plateformes, même non-Unix, mais aussi au dynamisme de l'équipe qui le maintient (Karl Berry, Thomas Esser, Ulrik Vieth, Wayne Sullivan, Bernd Raichle pour n'en nommer que quelques uns).