

Cahiers **GUT** *enberg*

☞ HISTOIRE DE TEX SOUS DOS ET WINDOWS
À L'ÉCOLE NATIONALE SUPÉRIEURE DE
PHYSIQUE DE STRASBOURG

☞ Marc TORZYNSKI

Cahiers GUTenberg, n° 25 (1996), p. 41-56.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1996__25_41_0>

© Association GUTenberg, 1996, 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.

Histoire de T_EX sous Dos et Windows à l'École nationale supérieure de physique de Strasbourg

Marc Torzynski

École nationale supérieure de physique de Strasbourg

Parc d'innovation

F-67400 ILLKIRCH

Marc.Torzynski@ensps.u-strasbg.fr

Télécopie : 88 65 52 49

Résumé. À travers une évocation de l'évolution de T_EX dans son institution, l'auteur présente comment il a été amené à développer différents environnements de travail sous Dos et Windows. Ces outils ont été installés sur le réseau interne de l'établissement et sont accessibles à l'ensemble des personnels et des étudiants.

1. La vie avant T_EX

L'École nationale supérieure de physique de Strasbourg (ENS_{PS}) était, il y a une dizaine d'années, l'une des plus petites des Écoles nationales supérieures d'ingénieurs et formait annuellement une quarantaine d'ingénieurs. Les documents des treize enseignants-chercheurs étaient généralement composés à l'aide de machines à marguerites interchangeables et les étudiants rédigeaient leurs rapports de stage... comme ils le pouvaient (c'est-à-dire essentiellement à la plume!).

Pour ma part, j'étais en 1987 la plus jeune recrue du corps enseignant, pas encore titulaire de surcroît. Je ne disposais donc pas encore de l'aura nécessaire pour convaincre notre unique secrétaire de bien vouloir taper mes textes d'examens et encore moins mes cours, aussi avais-je dû persuader mon directeur de laboratoire qu'un traitement de textes scientifique capable de tourner sur les PC 286 récemment livrés serait le bienvenu. C'est ainsi que j'ai gagné mon indépendance dactylographique avec un produit nommé MATHOR. La composition des formules devenait désormais un jeu d'enfant : je crois me souvenir qu'il suffisait de taper F2 pour insérer dans le texte un « $\sqrt{\square}$ » qui s'ajusterait ensuite aux dimensions du radicande que l'on substituerait au « \square » générique ; F6 faisait apparaître un \sum_{\square}^{\square} à compléter par les bornes de sommation désirées, et ainsi de suite.

Bien sûr, MATHOR avait quelques inconvénients. Le choix des tailles et styles de caractères était assez restreint. Il fallait d'ailleurs se limiter à n'utiliser que des polices à chasse fixe, car les formules sortaient très mal en mode proportionnel. Les imprimantes à laser n'ont été supportées que dans les dernières versions du programme (ce qui à vrai dire nous importait peu, puisque nous ne disposions de toutes façons que d'imprimantes à aiguilles). Enfin, à 5000 francs le jeu de disquettes verrouillées par une clé de protection, il n'était pas possible de multiplier les postes de travail et j'avais fort à faire pour me défendre des envieux.

2. Apparition de T_EX à l'ENS_{SPS}

Par le hasard d'une rencontre, je découvris un jour qu'à l'institut de recherches nucléaires du nord de la ville, il était possible de composer aussi bien des *preprints* de physique des ions lourds que des rapports de cristallographie biologique truffés d'intégrales et de graphiques, et avec une typographie digne de celle des meilleurs ouvrages de notre bibliothèque. J'appris que cette qualité était à mettre au compte d'un programme accessible à tout utilisateur du centre de calcul de cet institut. Je savais déjà que nous disposions d'un terminal, qui par chance servait très peu : je m'empressai donc de demander l'ouverture d'un compte, et c'est ainsi que T_EX fut utilisé pour la première fois à la fin de 1988 à l'ENS_{SPS}.

Les débuts furent laborieux : il fallait apprendre à maîtriser simultanément l'environnement général de travail MVS, l'interface ISPF, les principes de fonctionnement et la syntaxe de T_EX. Tout cela n'était pas facile car étant à l'ENS_{SPS} l'un des seuls utilisateurs, je ne pouvais guère espérer trouver de l'aide auprès de mes collègues ; quant à la documentation, elle se trouvait sur le site central, à 8 kilomètres... Heureusement, les ingénieurs système avaient bien fait les choses : aide à l'utilisation de T_EX et brochures d'auto-formation étaient disponibles en ligne. Grâce à celles-ci, je réussis assez rapidement à composer mes premiers documents. Je me souviens que le premier consistait en la page de titre de ma thèse, exercice périlleux pour un débutant puisqu'il fallait ajuster des `\hbox` dans des `\vtop`. Mais la qualité obtenue m'a toujours largement dédommagé des efforts engagés (celui de la traversée du campus à pied et en plein hiver pour récupérer les pages envoyées sur l'imprimante laser la plus proche n'était pas le moindre).

3. Découverte de T_EX sur PC

Après deux ou trois semestres d'utilisation et à force d'une propagande enthousiaste, j'avais réussi à faire quelques émules mais craignais désormais pour la disponibilité du terminal. Heureusement, un étudiant est arrivé un jour dans mon bureau pour y dé-

poser triomphalement un jeu de disquettes étiquetées « T_EX – version Ms-Dos ». Nous les avons fébrilement décompactées pour y trouver, en vrac, un moteur `sbtex`, une visionneuse `cdvi` et un pilote pour imprimante LaserJet que nous nous sommes empressé d'essayer. Terminée, l'attente de la fin du « traitement batch » et l'ouverture du fichier `log`, remplacée par la compilation en mode interactif... Oubliée, la peu conviviale visualisation par quart de page, place à l'agrandissement et la réduction – presque – à volonté du texte... Mais le plus important a été de découvrir dans la documentation qu'il existait un peu partout dans le monde des groupes d'utilisateurs de T_EX, des listes de discussion, des serveurs `ftp`. Voilà une information qui tombait à point nommé, puisque j'avais depuis quelques semaines accès à Internet depuis mon PC via Osiris (le réseau de notre université). L'exploration des ressources T_EX sur les réseaux a ainsi constitué mon occupation principale pendant une ou deux semaines.

4. Installation de T_EX sur réseau Ms-Dos

4.1. Une première tentative avortée

À la rentrée de septembre 1990, j'ai reçu la visite de deux étudiants. Revenant d'un stage effectué aux États-Unis, ils n'avaient pas voulu s'alourdir inutilement et n'avaient donc ramené de leur rapport que les sources T_EX qu'il s'agissait désormais d'imprimer. J'étais un peu embarrassé, puisque la seule solution que j'avais à leur offrir était de leur laisser l'accès à mon PC à l'heure du déjeuner ou en fin de journée. J'en vins alors à envisager une installation publique de T_EX sur le serveur de fichiers HP9000/615 desservant la quinzaine de PC 386 de la salle d'informatique récemment aménagée. Mais il ne pouvait être question de simplement recopier le contenu des répertoires de mon ordinateur sur le serveur. Je désirais en effet proposer une installation complète, conviviale, documentée, facilement utilisable pour un novice ; bref, de mettre à disposition des étudiants et des collègues l'équivalent de ce qui m'avait été offert sur le terminal du centre de calcul.

Je repris contact avec l'étudiant qui m'avait procuré les disquettes T_EX/Dos et lui proposai de travailler ensemble sur la question. Il s'avéra que lui-même y avait déjà réfléchi et me présenta un premier jet d'une interface utilisateur « à la Borland » développée avec TurboVision, destinée à gérer l'édition, la compilation et l'impression d'un texte. Nous nous sommes cependant vite heurtés à des problèmes de passage de variables d'environnement, de gestion de la mémoire, et n'avions ni l'envie ni le temps de poursuivre un travail qui s'avérait finalement trop ambitieux pour nous (mais heureusement pas pour d'autres, comme en témoigne `TeXshell` apparu un peu plus tard).

4.2. Une installation moderne, conviviale et francisée

Parallèlement, je commençai à regretter les limitations de ma visionneuse, l'absence de pilotes pour nos imprimantes à aiguilles et m'informai sur la liste de discussion Gutenberg de l'éventuelle existence d'outils plus performants que ceux dont je disposais. Un membre de la liste bien connu pour son travail sur la typographie musicale automatique me répondit qu'il disposait de la réponse à mes attentes et me pria de lui faire parvenir six disquettes vierges, lesquelles me revinrent d'Orsay quelques jours plus tard avec un petit mot indiquant simplement de « 1^o faire `pkunzi p -d` sur toutes les disquettes ; 2^o lire les docs ; 3^o [le] contacter si problème ». Je n'ai jamais eu besoin de le faire : les disquettes contenaient l'excellente et récente distribution `emTeX`, produit du domaine public devenu depuis le standard `TEX` sous Ms-Dos.

Muni de cet ensemble d'exécutables qui rendait obsolètes mes premiers outils `TEX` pour PC, il ne me restait plus qu'à créer un « enrobage » qui rende son utilisation transparente, indépendante des changements futurs, et surtout la plus simple possible. En effet, il faut bien avouer qu'un système « minimal » n'est guère attrayant. Certes, il est facile de faire admettre à un débutant que `TEX` est un système où la production des documents s'effectue en trois phases successives d'édition, de compilation et de visualisation/impression. Mais passée l'étape d'édition déjà considérée comme rébarbative, il lui devient vite fastidieux de devoir taper et retaper – n'oublions pas qu'il est novice et que les allers-retours édition/compilation sont nombreux – le nom des exécutables. Au moment de l'impression, il lui faudra encore se souvenir que les pilotes sont spécifiques à chaque imprimante, puis s'enquérir du nom des options (rarement les mêmes d'un pilote à l'autre) à rajouter sur la ligne de commande. Bref, de quoi en décourager plus d'un.

Puisque la production d'un document `TEX` est une tâche séquentielle, la forme que devait prendre un environnement facile d'utilisation s'est imposée naturellement : pourquoi ne pas présenter à l'utilisateur une succession de menus qui lui proposeraient au fur et à mesure les opérations à effectuer ? Et après tout, pourquoi ne pas faire appel à un simple fichier de commandes Ms-Dos à cet effet ? Il suffirait de présenter à l'utilisateur des écrans successifs pour guider ses pas, et de récupérer ses choix dans une variable d'environnement réservée à cet effet¹ ; l'aiguillage à travers l'arborescence des menus serait effectuée à l'aide des commandes conditionnelles de l'interpréteur de commandes ; enfin, à l'extrémité de chaque branche, il suffirait d'exécuter le programme adéquat de la distribution `emTeX` accompagné de ses arguments, puis de revenir au menu principal.

La structure générale a été rapidement définie, et j'ai disposé peu de temps après d'un environnement de travail opérationnel qui m'évitait le lancement fastidieux des pro-

1. Cette opération qui n'était pas supportée par Ms-Dos est très bien effectuée par divers petits utilitaires du domaine public – `what.exe` par exemple.

grammes à partir de la ligne de commande : il suffisait de sélectionner dans un menu affiché à l'écran l'opération à effectuer (édition par appui sur la touche E, compilation par C, etc.). Les petites mises au point, comme par exemple le positionnement précis de symboles nécessitant de fréquents allers/retours entre édition et visualisation, s'effectuent ainsi en un temps record.

J'avais cependant encore un point important à résoudre : jusqu'à présent, je n'avais guère abordé la question de la création des formats T_EX et de la francisation de ceux-ci, ne serait-ce que pour permettre la saisie et l'impression des lettres accentuées ainsi que la coupure des mots en contenant. J'ai donc dû me plonger dans les mystères des motifs de césure et des caractères actifs (on verra plus loin que ce n'était pas la dernière fois), domaine qui m'apparaissait plutôt hermétique. Heureusement, un petit livre écrit par un mathématicien strasbourgeois m'a éclairé sur bien des points. J'y ai également trouvé de nombreuses informations sur la francisation de la typographie, de sorte que j'ai rapidement obtenu, entre autres, une disposition et une gestion correcte des blancs avant et après les signes de ponctuation double.

Il ne me restait désormais plus qu'à m'occuper de la rédaction du *Local Guide*. D'une vingtaine de pages, sa première version contenait une introduction à T_EX, une description des outils utilisés, le manuel d'utilisation de l'interface utilisateur, des conseils pour une bonne typographie et diverses informations pratiques (coordonnées des listes de discussion, des différents serveurs, etc.) ; une seconde brochure constituait le catalogue des fontes disponibles. Finalement, ce n'est qu'en novembre 1991 que les fichiers ont été rendus accessibles aux utilisateurs des quinze PC en libre-service reliés au serveur HP9000 : il m'avait donc fallu environ un an pour finaliser mon idée d'une installation publique de T_EX à l'ENSPS et créer un environnement de travail complet guidé par menus pour rendre transparent le lancement des différents exécutable.

4.3. Les évolutions

Dans les deux années qui suivirent, j'installais de nouveaux outils et les rendais accessibles par les menus de l'interface. À la configuration initiale (T_EX, L^AT_EX, BibT_EX, pilotes écran, LaserJet et Nec 24 aiguilles) sont venus s'ajouter un correcteur orthographique (une avant-première d'amSpEl1, la version officielle posant problème avec les caractères accentués), des outils de dessin (T_EXcad), d'impression PostScript (dvi_{PS}), de gestion bibliographique (Bi b_db, là aussi une β-version corrigeant divers problèmes en 8 bits).

Il fallait aussi satisfaire les demandes des utilisateurs. Parfois simples, lorsqu'il s'agissait de la possibilité de choisir par menu son imprimante de sortie ou l'orientation du papier ; parfois plus délicates, comme la création d'un style local permettant l'impression de lettres avec le logotype de l'ENSPS. Plus tard, tgrind a été compilé pour Ms-Dos sous la pression des informaticiens, bm2font installé pour nos chercheurs en image-

rie, suivi de `gnuplot` pour la production des graphes des expérimentateurs. Enfin, l'abandon de mon pauvre style `frtypo` pour un style `french` bien plus performant a été l'occasion – collaborations transfrontalières oblige – de créer un style `allemand` compatible avec ce dernier, bien que de nombreux échanges de messages avec l'auteur du style et quelques longues soirées m'aient été nécessaires pour réussir à enfin associer règles de typographie allemande avec les possibilités de multilinguisme du style français.

Vers la fin de 1993, l'installation pouvait être considérée comme désormais pleinement opérationnelle et sa forme a été figée. Un menu principal (fig. 1) regroupe l'ensemble des opérations de base sur un fichier ne contenant que du texte. Des menus secondaires donnent accès à la gestion bibliographique, à l'éditeur graphique, aux outils de conversion et au paramétrage de l'impression. Enfin, pour améliorer la convivialité, les opérations proposées par chaque menu sont contextuelles : par exemple, visualisation et impression ne sont pas accessibles si le fichier `dvi` n'est pas présent sur disque. À l'usage, j'avais de moins en moins de messages m'informant de tel problème, ou me demandant s'il « était possible de... ». Par contre, les « j'ai copié l'installation `TEX` sur ma propre machine, merci de me tenir au courant des évolutions... » augmentaient. Mais je voyais approcher la date, prévue pour dans deux mois, du transfert de l'ENS_{PS} du centre-ville vers le campus sud, et je savais que ce serait l'occasion du remplacement de bon nombre de PC sous Ms-Dos par de plus puissants tournant sous Windows : si je voulais simplement conserver, à défaut d'en former de nouveaux, les adeptes de `TEX`, il fallait que j'envisage sérieusement une mise à niveau de mon interface de travail.

5. Migration de `TEX` vers Windows for Workgroups

5.1. La nouvelle situation

Le déménagement de l'ENS_{PS} en février 1994 nous fit passer de la vie familiale à la communauté : la surface des locaux passa de 2000 à 22 000 m² ; différentes équipes de recherche jusqu'alors dispersées se regroupèrent ; une école de biotechnologie vint nous rejoindre. Au total, la population des lieux s'établit à près de 500 personnes contre une petite centaine quelques mois plus tôt.

L'environnement informatique jusqu'ici plutôt pauvre en fut évidemment profondément bouleversé, grâce aussi à notre nouvel ingénieur système. À côté de système Unix davantage destinés aux travaux de recherche, un réseau d'usage général (messagerie, bureautique) Windows for Workgroups géré par deux serveurs Dec xLS fut très rapidement opérationnel et potentiellement accessible depuis le moindre recoin du bâtiment grâce au précâblage des lieux. Ce sont ainsi plus de 200 PC qui ont été interconnectés en moins de trois mois.

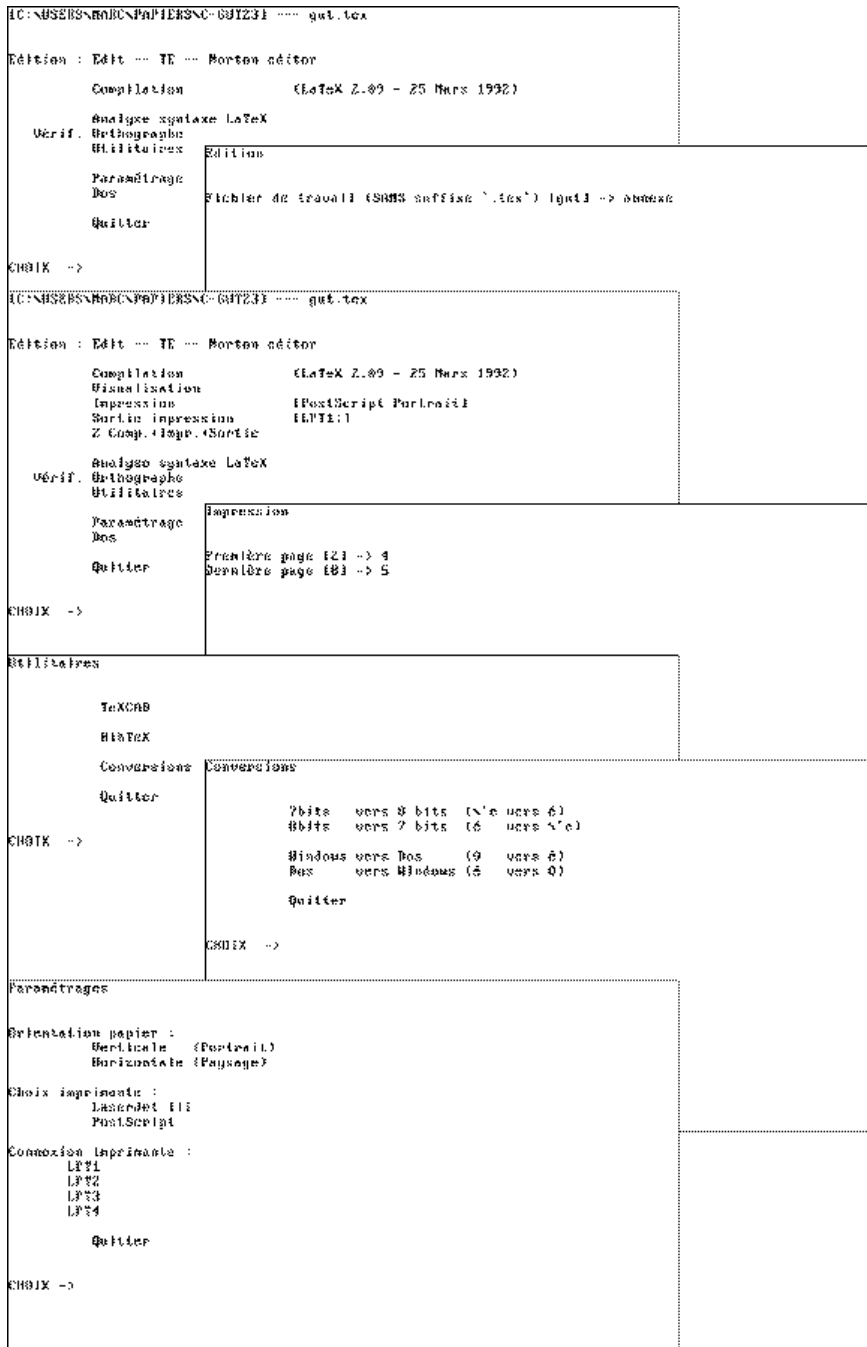


FIGURE 1 – Différents menus de l'environnement de travail $\text{T}_\text{E}\text{X}$ sous Ms-Dos. Observer comment le contenu du menu principal évolue selon l'état d'avancement du travail.

Si je reconnaissais à l'usage que le choix d'un réseau Windows n'était finalement pas si mauvais, je n'adhérais bien entendu pas du tout à celui d'imposer de fait l'utilisation de Microsoft Office. Non pas par *a priori* ; mais mes deux seules expériences avec ce produit s'étaient avérées absolument décevantes. La première, lorsque j'ai essayé des incompatibilités entre fichiers créés sur des systèmes différents. La seconde, alors que j'avais demandé aux différents auteurs d'un ouvrage collectif de me remettre leurs textes au « format Word », pensant que je n'aurais qu'à uniformiser les polices de caractères pour disposer d'un ensemble homogène ; hélas ! je me suis vite aperçu que chaque disquette nécessitait un éditeur graphique différent pour la modification des schémas, et qu'il en allait de même pour les équations mathématiques. (Bien entendu, les auteurs étaient tous persuadés de n'avoir utilisé que des outils parfaitement standards et classiques.)

Il me fallait donc trouver une solution pour mettre $\text{T}_{\text{E}}\text{X}$ à la disposition de cette catégorie d'utilisateurs pour qui icône, souris et menu déroulant constituaient désormais la Sainte Trinité informatique et avaient oublié jusqu'à l'existence d'une invite de commande. Malheureusement, il n'existait rien permettant d'interfacer $\text{T}_{\text{E}}\text{X}$ avec Windows, à l'exception du pilote d'écran `dviwin`. Si l'environnement développé pour Ms-Dos restait d'actualité pour les « anciens » (pour ma part, je persiste à l'utiliser pour toutes les mises au point et autres travaux nécessitant des cycles compilation/visualisation répétés, en raison de la rapidité de manipulation et d'exécution qu'il permet), il m'était impossible de le proposer aux nouveaux arrivants.

Évidemment, ayant abandonné le développement d'un environnement souris/menus déroulants sous Ms-Dos, je n'allais pas m'attaquer à un travail analogue sous Windows. Mais un stagiaire de notre équipe m'avait parlé d'un éditeur qu'il était possible de paramétrer pour pouvoir (par exemple) lancer depuis celui-ci la compilation C d'un fichier en cours d'édition. Compilation C... et pourquoi pas une compilation $\text{T}_{\text{E}}\text{X}$? Je me procurai donc dans le domaine public cet éditeur programmable, en l'occurrence la version Windows de Micro-Emacs.

5.2. $\text{T}_{\text{E}}\text{X}$ et Micro-Emacs pour Windows

La fenêtre de Micro-Emacs pour Windows s'apparente à celle de n'importe quelle application Windows : menus déroulants, boîtes de dialogue, curseurs en bordure de fenêtre, aide en ligne (bien entendu, les commandes clavier standard Emacs restent opérationnelles). Les fonctionnalités de base sont celles de tout éditeur multi-fenêtres, mais le langage de programmation permet sa reconfiguration grâce aux possibilités de suppression/création/redéfinition des menus et de leur contenu, d'ouverture de boîtes de dialogue pour la sélection des fichiers, de lecture des entrées clavier, de lancement d'exécutables ou de commandes Ms-Dos ou Windows avec ou sans passage d'arguments, en avant ou arrière-plan. La personnalisation peut être d'autant plus poussée qu'Emacs dispose d'un jeu complet de directives d'exécution séquentielle, condition-

nelle ou itérative pour l'écriture de procédures complexes qu'il est ensuite possible d'appeler à partir des menus.

Fort de cet éditeur, je n'eus pas de grandes difficultés à imaginer la réalisation d'un nouvel environnement de travail. Il s'agissait simplement d'étendre à Windows les principes utilisés pour l'environnement MS-Dos, donc d'utiliser les macro-commandes d'Emacs pour créer une interface permettant d'accéder à l'ensemble du corpus d'exécutables associés à \TeX . La différence résidait simplement dans le fait que l'interface était désormais constituée par l'éditeur lui-même. Cette différence est malgré tout de taille, puisque grâce à la conjonction des fonctionnalités multitâches de Windows et d'Emacs, les opérations conduisant à la production proprement dite du document (compilations/visualisations répétées pour les petites mises au point, impressions d'épreuves, etc.) pourraient être effectuées sans quitter le(s) fichier(s) de travail. Bien entendu, ces opérations seraient confiées aux mêmes outils que ceux déjà utilisés sous Ms-Dos, de façon à ne pas modifier l'arborescence de fichiers existante et à préserver l'utilisation de l'environnement Dos. La configuration d'Emacs (c'est-à-dire l'écriture des procédures dans un fichier « `rc` » et la création des fichiers permettant le lancement des différents exécutables) s'est alors effectuée par étapes successives, en suivant la progression des tâches nécessaires à la production d'un document \TeX et en créant à chaque fois les procédures et menus nécessaires. La description ci-dessous de l'environnement de travail ainsi créé suit le même chemin.

5.3. Déroulement d'une session \TeX sous Windows

De prime abord, Emacs apparaît légèrement appauvri : dans un souci de simplification, l'arborescence de base des menus a été quelque peu élaguée (réduction du contenu des menus d'édition, de recherche et de gestion des fenêtres). Mais surtout, le menu « `File` » introduit la notion de fichier « racine » ou « secondaire ». En effet, puisqu'Emacs est un éditeur multi-fenêtres, il convient de discerner le fichier principal (celui sur lequel opérera le compilateur \TeX) des fichiers annexes (fichier `\include`, notes de travail). Concrètement (fig. 2, première fenêtre), l'utilisateur dispose de deux fonctions pour ouvrir un fichier. L'ouverture de nature « secondaire » est classique : une fenêtre de dialogue apparaît, le nom du fichier est saisi au clavier ou sélectionné interactivement dans une liste, puis son contenu est affiché dans une nouvelle fenêtre Emacs créée à son intention (la fenêtre est évidemment vide s'il s'agit d'une création de fichier). L'ouverture d'un fichier « racine » est particulière et diffère de la précédente sur deux points : 1° l'utilisateur doit spécifier quel est le format de travail associé à ce fichier (\TeX , \LaTeX ...), 2° si le fichier est vide (création), Emacs insère automatiquement quelques lignes de préambule standard selon le format de travail (`\documentclass[a4paper]{article}`, `\usepackage{french}`, `\begin{document}`, etc. pour \LaTeX). L'utilisateur n'a ensuite plus qu'à procéder à l'édition de son fichier en s'aidant des outils offerts par Emacs.

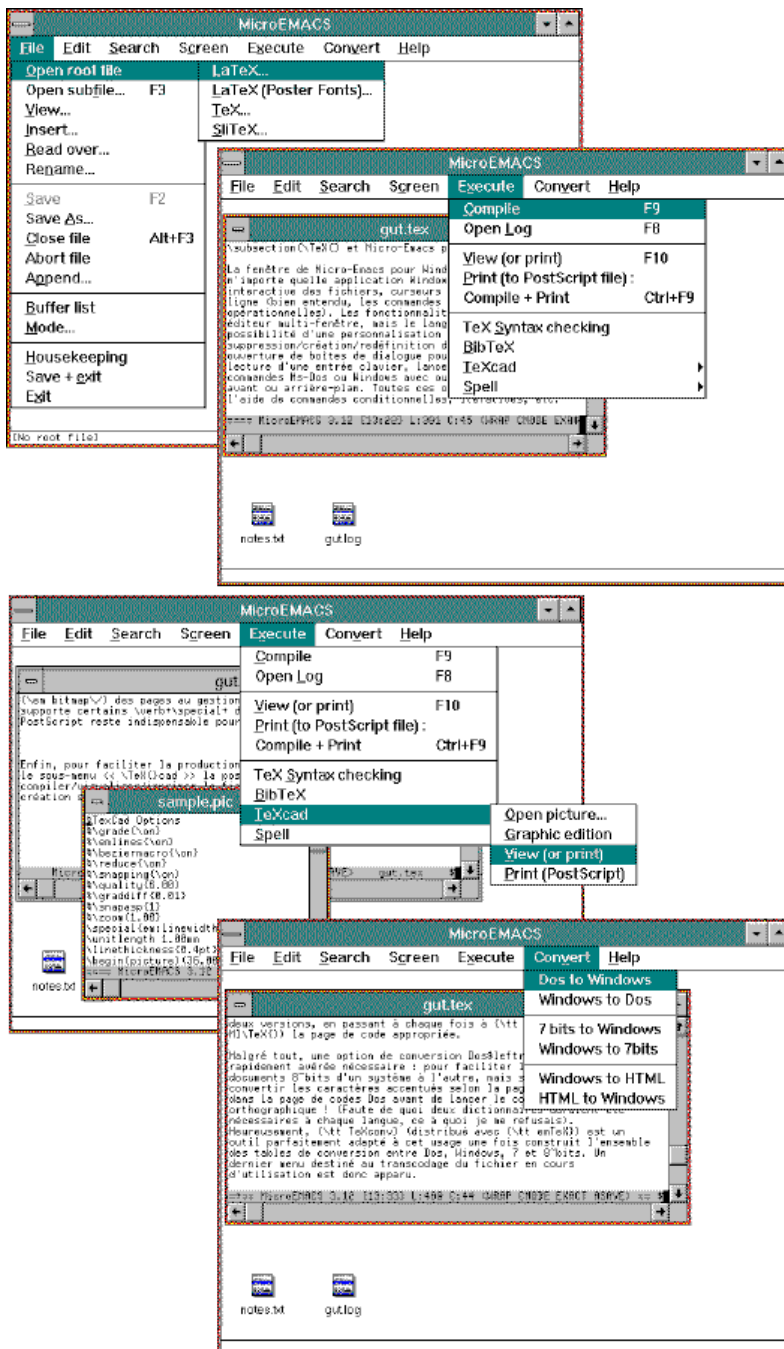


FIGURE 2 – Quatre aspects de l'environnement de travail \TeX sous Windows. Seule la fenêtre principale est représentée, bien que chaque opération (compilation, visualisation, etc.) soit susceptible d'ouvrir des fenêtres secondaires.

Cette tâche accomplie, il convient de procéder à la succession des opérations de traitement \TeX du fichier, accessibles par un menu « `Execute` » développé à cet effet (fig. 2, deuxième fenêtre). La première de ces opérations est celle de compilation qui, quelle que soit la fenêtre courante, opère toujours sur le fichier désigné comme « racine ». Ce fichier est éventuellement rappelé à l'écran s'il a été fermé, puis sa compilation (`emTeX` avec extensions 32 bits) est lancée dans une fenêtre Ms-Dos. La compilation est bien entendue interactive, mais si celle-ci se prolonge, rien n'empêche l'utilisateur de faire passer la fenêtre de compilation en arrière-plan afin de poursuivre son travail avec Emacs (à charge pour lui d'insérer dans son fichier \TeX un `\scrollmode` ou un `\batchmode` pour forcer \TeX à poursuivre le travail sans attendre de réponse clavier). La compilation effectuée, l'utilisateur a la possibilité d'ouvrir le rapport de compilation (ce qui n'est pas nécessaire si une fenêtre est déjà ouverte sur le fichier « `log` », puisqu'elle est automatiquement réactualisée dans ce cas).

La visualisation du fichier est effectuée par un appel à `dviwin`. Ce pilote d'écran opère sous Windows, de sorte qu'il n'y a aucun problème de cohabitation avec Emacs : la fenêtre de visualisation et les actions qui s'y déroulent sont totalement indépendantes de celles d'Emacs. L'utilisateur peut ainsi sans difficulté passer de l'une à l'autre pour procéder dans le source aux corrections rendues nécessaires par l'observation du `dvi`. Comme par ailleurs `dviwin` scrute les modifications du fichier `dvi` pour actualiser automatiquement le contenu de sa fenêtre si nécessaire, son association avec l'édition et la compilation sous Emacs permet un travail presque en temps réel.

`Dviwin` gère simultanément les fontes aussi bien individuelles (`pk`) que compactées en librairies (`fli`), et peut les rechercher dans des répertoires multiples. Un autre intérêt de `dviwin` est d'accepter bon nombre de `\special` (ceux de `tpic` et d'`emTeX` en particulier), ce qui permet un éprouvage sur écran plus efficace. Ce dernier point, associé à la possibilité de `dviwin` de pouvoir imprimer le fichier `dvi` sur toute imprimante gérée par Windows, évite l'installation de pilotes d'imprimante multiples. Cependant, il existe un gros inconvénient lié à l'envoi par `dviwin` au pilote d'impression Windows des pages sous forme de plans de bits (*bitmap*) : les temps d'impression et les exigences de mémoire augmentent dangereusement au delà d'une résolution de 300 points par pouce (DPI).

Pour pallier cet inconvénient, de taille aujourd'hui puisqu'une résolution de 600 DPI est courante sur les imprimantes actuelles, il convenait de disposer malgré tout d'un « vrai » pilote d'imprimante. Une règle tacite à l'`ENSPS` voulant que tout échange de document sous forme électronique se fasse soit en texte « pur », soit en PostScript, c'est `dvips` qui a été retenu. L'exécution de ce pilote se déroule de la même façon que la compilation, c'est-à-dire dans une fenêtre Ms-Dos qui laisse à l'utilisateur la possibilité de poursuivre son travail pendant que s'effectue la conversion du fichier `dvi` en PostScript. (C'est d'ailleurs ici que le travail en arrière-plan trouve son véritable intérêt : si pour du texte pur le transcodage `dvi` → `ps` est comparable à la composition `tex` → `dvi`, l'inclusion par `dvips` sous Windows d'une image de taille A4 peut-être près de 20 fois

plus lente que le traitement d'une page de texte de même format.) L'impression du fichier PostScript se fait ensuite à l'aide de `prfle`, petit utilitaire Windows du domaine public permettant l'envoi direct de fichiers vers une imprimante (opération que Windows ne permet pas).

Outre les fonctions indispensables décrites précédemment, la personnalisation du menu « `Execute` » d'Emacs permet également l'accès à divers outils classiques : appel à BibTeX, correction orthographique et vérification de la syntaxe TeX (par appel au programme `texchk` de la distribution emTeX, particulièrement utile pour détecter les non-appariements de délimiteurs de blocs). Il s'agit toujours d'outils exécutés dans des fenêtres Ms-Dos, mais si BibTeX opère bien évidemment sur le fichier aux relatif au fichier « racine », l'exécution des deux autres programmes peut porter sur n'importe lequel des fichiers ouverts (en l'occurrence celui figurant dans la fenêtre courante).

Le dernier outil accessible par le menu est constitué par TeXcad (un éditeur de schémas permettant la création interactive graphique d'environnements L^AT_EX de type `picture`), pour lequel l'environnement de travail propose une utilisation un peu particulière. Les idées sous-jacentes étaient d'une part de rendre la mise au point du schéma indépendante de son inclusion dans le document d'accueil, d'autre part d'accéder facilement au code `picture` pour le modifier manuellement, enfin de permettre la création et l'impression de schémas utilisant les polices TeX sans nécessairement devoir les inclure dans un document (ou sans le pouvoir, cas des utilisateurs de TeX natif).

Pour ces raisons, le travail sur un schéma s'effectue en plusieurs étapes (fig. 2, troisième fenêtre). La première est la sélection du fichier « `picture` » sur lequel on souhaite travailler, opération qui ouvre une fenêtre Emacs (vide ou pas selon l'existence préalable du fichier). Cette fenêtre permet la création/modification manuelle du fichier, mais l'utilisateur peut à ce moment opter pour une édition graphique grâce à l'exécution de TeXcad dans une fenêtre Ms-Dos plein écran. L'édition graphique achevée, la fenêtre Emacs associée au fichier `picture` est actualisée. À ce moment, il est possible de demander une visualisation ou une impression du seul schéma : de façon transparente pour l'utilisateur, l'environnement `picture` est inséré dans un fichier L^AT_EX « minimal » puis compilé, visualisé ou imprimé. La mise au point d'un schéma peut ainsi se faire par cycles successifs d'édition/compilation/visualisation totalement indépendants du document d'accueil, bien que simultanés avec l'édition de celui-ci.

5.4. Comment ça marche?

Il reste à expliquer comment un simple appel à la souris sur (par exemple) l'article « `Create PS` » permet de lancer l'impression par `dvips` des pages 3 à 5 du document de travail. Signalons d'abord que lors de son lancement, Emacs lit dans des fichiers « `rc` » un ensemble d'instructions permettant son auto-configuration et stocke un ensemble de procédures (à la manière de TeX qui en lisant un format initialise des tables

internes avec des motifs de césure et stocke un ensemble de macros). Certaines de ces procédures (celles du fichier `emacs.rc`) sont propres à Emacs, d'autres sont précisément celles que l'utilisateur a développées pour ses besoins propres. Par exemple, une procédure a été associée à l'ouverture du fichier « racine », qui permet la séparation et le stockage dans différentes variables du nom du fichier proprement dit, de son extension et du nom du répertoire de travail.

Ainsi, l'appel à l'article « `Create PS` » lance la procédure qui lui est associée. Cette procédure commence, sur la base des variables stockées précédemment, par construire le nom du fichier `dvi` à convertir, puis teste son existence. En cas d'absence de celui-ci, un message le signale et invite l'utilisateur à d'abord compiler le fichier source, puis la procédure se termine. (Il n'est pas possible de modifier dynamiquement le contenu des menus, aussi faut-il se prémunir contre les mises en place de la charrue avant les bœufs.) Dans le cas contraire, elle se poursuit en demandant au moyen de la ligne de messages d'Emacs de donner les numéros des première et dernière pages à imprimer, et récupère ainsi deux entrées clavier fournies sur cette même ligne. La procédure construit ensuite la commande complète à exécuter (programme et arguments) par concaténation de variables statiques (par ex. la chaîne spécifiant le chemin complet de la commande de lancement de `dvips` sans argument) et dynamiques (par ex. le nom du fichier `dvi` ou les numéros des pages concernées).

La procédure n'a plus qu'à demander l'exécution de la commande après avoir précisé le mode adéquat : il est « non-synchronisé » dans le cas présent, puisque l'on souhaite que l'impression puisse s'effectuer en arrière-plan et qu'un mode « synchronisé » imposerait au contraire d'attendre la fin de l'impression avant de poursuivre l'édition. `Dvips` n'étant pas une application Windows, Emacs ouvre automatiquement une fenêtre Ms-Dos pour l'accueil de son exécution. Celle-ci terminée, la fenêtre est refermée et les opérations peuvent se poursuivre.

Précisons pour finir la façon particulière dont l'exécution est lancée. En effet, la commande lancée par Emacs ne contenait pas le mot `dvips32.exe`, bien que ce soit effectivement le nom du programme exécuté. En réalité, Emacs a lancé un fichier `dvips.pif` auquel il a passé les arguments nécessaires à `dvips32.exe` : procéder ainsi permet de contrôler pour chaque application la fraction du temps CPU qui lui est affectée par Windows. Le fichier `pif` ne lancera ensuite qu'un simple fichier `dvips.bat`, de façon à effectuer le positionnement de diverses variables d'environnement avant le lancement de `dvips32.exe`. À l'usage, il est apparu que la configuration judicieuse des paramètres d'exécution dans les différents fichiers `pif` conditionne grandement le fonctionnement optimal de l'environnement.

5.5. À propos des accents

Parallèlement à la création de l'environnement basé sur Emacs, la question des caractères accentués a refait son apparition lorsque je me suis souvenu que Ms-Dos et Windows n'utilisaient pas les mêmes tables de caractères. Sous Dos, l'activation des caractères et l'utilisation de fontes virtuelles avaient permis de gérer de façon satisfaisante l'accentuation des lettres et la coupure des mots. Il aurait été possible de concevoir un format unique pour Dos et Windows basé sur ce principe, puisque les jeux de caractères des deux systèmes ne se recouvrent pas (ou très peu). Mais l'utilisation des fontes virtuelles avait fini par m'apparaître peu satisfaisante. Aussi renonçai-je à l'activation des caractères pour utiliser le mode $\text{M}\text{T}\text{E}\text{X}$ d' $\text{e}\text{m}\text{T}\text{e}\text{X}$ et chaque format a finalement été généré en deux versions, l'une pour Dos, l'autre pour Windows.

De fait, une commande de conversion des caractères selon les tables Dos et Windows s'est rapidement avérée nécessaire. D'abord pour faciliter le portage des documents 8 bits d'un système à l'autre ; ensuite et surtout pour convertir les caractères accentués selon la page de codes Windows dans la page Dos avant de lancer le correcteur orthographique, faute de quoi il aurait fallu comme pour les formats disposer pour chaque langue de deux versions des dictionnaires, ce à quoi je me refusais. Heureusement, $\text{T}\text{e}\text{X}\text{conv}$ (distribué avec $\text{e}\text{m}\text{T}\text{e}\text{X}$) est un outil parfaitement adapté à cet usage une fois construit l'ensemble des tables de conversion entre Dos, Windows, 7 et 8 bits. Un dernier menu destiné au transcodage des caractères du fichier en cours d'utilisation est donc apparu (fig. 2, dernière fenêtre).

5.6. La mise en réseau

En octobre 1994, l'environnement TEX sous Windows était opérationnel et je proposai à l'ingénieur système son affichage dans un répertoire public des serveurs. Partisan d'une utilisation uniforme de Word au sein du site, celui-ci n'a pas fait montre d'un enthousiasme débordant. Mais c'était sans compter sur mes complicités au sein du service informatique : c'est finalement son assistante qui a créé un répertoire TEX sur le serveur, installé les fichiers, et m'en accordé les droits d'écriture pour terminer la mise au point.

Le principe de cette mise en réseau est simple. *Tous* les fichiers nécessaires à TEX résident sur un disque du serveur, y compris le fichier de groupe pour Windows qui constitue le point d'entrée de l'environnement. Puisque l'ensemble des outils relatifs à TEX est accessibles à partir de l'éditeur, ce fichier de groupe ne contient que trois icônes : une pour appeler Emacs, deux pointant vers les fichiers de documentation (informations de première nécessité et *Local Guide*). À leur tour, Emacs et les fichiers pif ou bat associés font référence au disque du serveur pour la localisation des exécutables.

L'utilisation de T_EX par un utilisateur distant est ainsi extrêmement aisée et « économique », puisqu'il ne lui est pas nécessaire d'installer le moindre fichier sur sa machine. Même la présence du fichier de groupe est superflue, puisque Windows se contente de la *référence* à ce fichier. La seule contrainte est de devoir « monter » le disque du serveur sous un nom imposé. Ce nom est en effet codé en dur dans les fichiers car Emacs ne prend pas en compte les variables d'environnement dans son fichier de configuration.

Les premiers essais ont réservé l'agréable surprise de constater que même sur un réseau de cette importance géré par des machines de puissance limitée, les temps de réponse restent parfaitement acceptables. D'ailleurs, une fois les paramètres d'exécution convenablement ajustés dans les fichiers `pi f` (gestion des priorités et répartition du temps d'exécution), le travail à partir d'un disque distant est totalement transparent pour l'utilisateur.

Après quelques semaines de mise au point et de tests intensifs effectués par des collègues de bonne volonté, j'envoyais une annonce à l'ensemble des utilisateurs du réseau pour les informer de la disponibilité de T_EX sous Windows. Un peu plus tard, quelques affiches de promotion sont apparues autour des salles d'informatique à accès libre, suivies de l'acquisition par la bibliothèque de divers ouvrages de référence sur T_EX, L^AT_EX ou la typographie...

6. En guise de conclusion

Après un an et demi de fonctionnement sous Windows, l'essentiel des problèmes semble avoir été détecté et corrigé, et l'environnement fonctionne de façon stable. De l'avis des utilisateurs, il s'avère fonctionnel et très agréable, et je sais qu'il est utilisé régulièrement ou occasionnellement aussi bien par des étudiants que des chercheurs, éventuellement à domicile. En effet, chaque PC du site peut accéder aux fichiers du serveur, aussi bien en exécution qu'en lecture pour une copie en vue d'une installation sur un disque local. Toutefois, en l'absence d'outils permettant le décompte, l'origine et la nature des accès aux répertoires partagés, il est impossible de procéder à des statistiques précises d'utilisation. L'estimation est donc partiellement subjective et basée sur la discussion directe, les sondages et l'évaluation de la production de documents internes : si seul un rapport de stage sur 10 est écrit en T_EX (proportion faible car de nombreux stages sont effectués en milieu industriel), près du tiers des thèses et la moitié des documents pédagogiques (fascicules de cours ou de travaux pratiques) le sont. Il y a donc bien une présence « officielle » de T_EX à l'ENS_{PS}, ce qui prouve, s'il en était besoin, qu'il existe une place à côté des standards de fait commercialement imposés.

Hormis la création d'un format, d'un style et des fontes nécessaires à la production de *posters* (nous disposons d'une imprimante couleur A0) et l'installation de Ghostview pour leur manipulation avant impression, l'installation n'a guère évolué cette dernière année (elle a été récemment testée avec succès sous Windows 95, au prix de quelques

modifications mineures). Sont cependant envisagées des initiations à $\text{T}_{\text{E}}\text{X}$ dans le cadre du club « Info&Rézo » associant étudiants et enseignants du site, la conversion automatique et la mise en accès public au format Bib $\text{T}_{\text{E}}\text{X}$ des fichiers OCLC au format MARK répertoriant le fonds bibliothécaire du site, la création de pages w^3 de présentation des outils $\text{T}_{\text{E}}\text{X}$ à l'ENSPS, l'installation d'un convertisseur $\text{T}_{\text{E}}\text{X} \rightarrow \text{HTML}$, etc. Les projets ne manquent pas...

7. Civilités

Je profite de cet article pour saluer au passage les divers protagonistes implicitement cités au long de cette évocation. Par ordre d'apparition : ALeX Bilwes et al., Georges Weil et Jacques Morel, Cyril Boucher, Daniel Taupin, Raymond Séroul, Erik Frambach, Eyal Doron, Bernard Gaulle, Olivier Rolland, Christophe Pallier, Stéphane Casset, Catherine Maillot.