Le French-HOWTO : Linux & la langue française

Écrit par Guylhem Aznar, guylhem à oeil.qc.ca v2.1 ; Juillet 1998

Copyright (©) 1997, 1998 Guylhem Aznar. Librement distribuable et modifiable selon les termes du LDP

Table des matières

1 Présentation

1.1 Introduction

Ce document a été écrit pour aider à configurer un ordinateur doté du système d'exploitation Linux pour des utilisateurs francophones ; il y est expliqué par exemple comment configurer périphériques et applications afin de prendre en compte les particularités et les spécificités de notre langue...

Toutefois, ce document ne traite pas de l'installation du système d'exploitation Linux ; il est implicitement supposé que vous avez correctement installé une distribution de Linux (RedHat, Debian ou Slackware, cette liste n'étant pas exhaustive) et que vous vous apprêtez maintenant à mieux la configurer ; veuillez donc vous référer à d'autres HOWTOs pour l'installation.

N'étant pas l'unique auteur de ce document, les personnes m'ayant données des conseils à rajouter sont explicitement citées.

Les adresses électroniques de ces contribueurs ont été volontairement modifiées en remplaçant les «a enroulés» par des «à» afin d'éviter les publipostages très gênants.

1.2 Les français ne sont pas les seuls francophones !

Pour chaque manipulation dépendant du pays concerné, quatres versions sont donc proposées : une canadienne française, une belge wallone, une française et une suisse romande. Le Grand Duché du Luxembourg utilisant les conventions françaises, j'invite les lecteurs luxembourgeois à se reporter aux exemples français !

Toutefois seuls les fichiers destinés aux français (table de clavier en mode console, X...) sont inclus, puisque la majorité des lecteurs seront certainement français et que seuls ces fichiers présentent des améliorations par rapport aux défauts.

Envoyez-moi donc vos fichiers belges, canadiens, luxembourgeois ou suisses ; ils seront inclus avec plaisir s'ils se montrent plus utiles que ceux proposés par défaut !

1.3 Attention

Nat Makarevitch nous recommande :

ROOT : certaines manipulations recommandées par ce document ne peuvent être effectuées que sous un compte privilégié (root) ; la plus grande prudence est alors de rigueur car toute action inadéquate peut mettre en péril le contenu des disques. Sauvegarder les fichiers existants, par exemple en les copiant en *.old, avant de les modifier ou remplacer.

CHERCHER : le paquetage GNU nommé "find" offre un programme appelé `locate`, grâce auquel l'utilisateur peut retrouver le nom complet (chemin de répertoires) d'un fichier dont il ne connait que le nom.

Certaines distributions Linux fournissent cela en /usr/lib/locate. Lancer, en tant que root, `/usr/lib/locate/updatedb` : cela invoque un `find /` sur les disques montés et place les noms de tous les fichiers (sous forme hash) dans /usr/lib/locate/find.codes alors que `locate` permet ensuite de localiser l'un d'eux :

```
(guylhem@barberouge:guylhem)# locate noms_des_fichiers_à_retrouver
```

2 Les problèmes

2.1 Présentation

Le problème de tout utilisateur non-américain lors de l'installation de Linux est de lui faire comprendre que, justement, il n'est pas américain !

Cela a des conséquences sur :

- les caractères spéciaux que l'on utilise, accentués (É È À Ç) ou bien ligaturés (Æ , voire «e dans l'o» qui vient à peine d'apparaître sous Linux en mode console ; j'espère bientôt présenter ici une solution simple pour en disposer sous X aussi !)
- la correspondance entre la sérigraphie des cabochons (touches) sur lesquels vous appuyez et les codes que Linux renvoie, celui-ci supposant un clavier QWERTY américain par défaut
- le format de page pris en charge par les programmes que vous utilisez pour imprimer, différent entre les États-Unis (14 x 8,5 pouces pour du papier dit légal) et l'Europe (29,7 x 21 cm pour des feuilles a4)
- le format des dates et de l'heure
- le format de la monnaie
- la langue utilisée pour les messages d'erreur

2.2 Solutions

Tout étant préconfiguré pour un anglophone nord-américain -bien que ceux-ci ne représentent pas la majorité de la population mondiale- , il faut donc se faire reconnaître comme une «exception» par les différents programmes, par des options ou des variables à exporter voire dans les pires des cas une modification du code source suivie d'une recompilation.

Heureusement, «l'esprit» de Linux rend le système très ouvert à des modifications de toute sorte et comme vous allez le découvrir au fil de ce document, il est assez simple d'obtenir un résultat très correct.

2.3 Aide

Pour le moment, le futur de l'internationalisation de Linux porte sur la traduction des messages d'erreur, des programmes (comme le KDE), des pages de manuel et de la documentation, alors si vous vous sentez l'âme d'un traducteur, n'hésitez pas à contacter l'équipe de traduction des HOWTOs ; pour celà, adressez-vous à Éric Dumas (dumas à freenix.fr ou dumas à linux.eu.org) .

3 Le vocabulaire : petit lexique

Tout d'abord, un mot sur le vocabulaire employé :

3.1 8 bits

Des caractères sont dits «8 bits» s'ils contiennent par exemple des accents ou des signes spéciaux (comme paragraphe), non inclus dans le standard `ASCII` 7 bits (sans accents ni caractères spéciaux) qui correspond aux 127 premiers caractères des 256.

3.2 Table de clavier

Une table de clavier est un fichier permettant de faire correspondre l'empreinte physique des cabochons (touches) du clavier au résultat obtenu sur l'écran de l'ordinateur.

3.3 «Home», «End», «Page Up» et «Page Down»

Les touches «Home», «End», «Page Up» et «Page Down» (dont les noms varient selon les claviers) sont les touches permettant respectivement d'aller en début/fin de ligne et de monter/descendre d'une page.

Par exemple, pour que les touches Home et End fonctionnent dans le shell bash (voir plus bas), on peut mettre dans un fichier .inputrc les lignes suivantes :

```
"\e[1~" : beginning-of-line
"\e[4~" : end-of-line
```

D'autre part, on sait que les codes affectes a ces touches sont respectivement 102 et 107 pour Home et End en mode terminal.

Le code est issu du clavier : lorsqu'on presse la touche, il le renvoit à la table de clavier qui, lorsqu'elle le reçoit, se dit : «tient, c'est la touche End qu'on a pressé, donc le renvoie la chaîne associée, en l'occurence pour End «\e[4 » »

Lorsque bash ou un autre programme reçoit cette chaîne... il ne la connaît pas ! Donc il faut lui dire dans le .inputrc que «\e[4 » signifie «aller à la fin de la ligne».

3.4 «Touches mortes»

Les touches mortes sont des touches qui ne produisent pas de caractères en elles-mêmes, mais qui modifient la façon dont la frappe suivante va être interprétée (par exemple, le tréma ou l'accent circonflexe sont en général des touches mortes.)

Une touche morte est une touche qui doit donc être suivie d'un espace pour obtenir l'accent seul, ou dans le cas où elle est suivie d'une voyelle y positionne un accent, par exemple :

```
^ e donne ê
```

Les touches mortes sont :

- pour les claviers belges et français, l'accent aigu (en AltGr de 1 pour les français et en AltGr de ù pour les belges), l'accent grave (en AltGr de 7 pour les français & en AltGr de carré/cube pour les belges), le tréma et l'accent circonflexe (tous deux à côté du P)
- pour les canadiens, les accent graves, circonflexes, les trémas et la cédille
- pour les suisses, les accent aigus, graves, circonflexes, les trémas et la tilde

3.5 «Backspace» , «Delete»

- BackSpace Touche placée en haut à droite du pavé principal, au dessus d' «Entrée», au cabochon marqué d'une flèche vers la gauche.
- Delete Touche du pavé curseur au cabochon marqué «Suppr» ou «Del», à gauche de «Fin». Le pavé numérique offre une touche au marquage identique et à effet identique lorsque «Verr Num» *NumLock est desactivé*.

Ces deux touches correspondent chacune à un de ces codes :

- BS Caractère ASCII 0x08, ou control-h.
- DEL Caractère ASCII 0x7F, ou control- ?
- ^D Caractère ASCII 0x04, ou control-d.

Ceci dans le but d'effacer à gauche du curseur avec Backspace, à droite du curseur avec Delete.

3.6 Police de caractères

Une police de caractères est une correspondance entre des codes informatiques ou nombres hexadécimaux et des caractères (alphanumériques...) affichables par l'ordinateur.

Cette correspondance se nomme «norme» ou «standard» (comme l'unicode, l'iso-8859 latin, le cp ...).

Par exemple, dans le standard ASCII étendu par l'iso-8859-1, le nombre hexadécimal «eb» correspond à «ë».

La police de caractère est la somme des représentations graphiques des caractères d'un standard.

3.7 Unicode, iso-8859, latin

L'unicode est l'un de ces standard, destiné à terme à remplacer les différentes évolutions de l'ASCII.

L'ASCII codait ses caratères sur 7bits, rendant donc impossible d'incorporer des caractères accentués à la norme déja présente.

Les normes iso-8859 latin et cp (pour les plus connues) vinrent pallier à cette déficience en encodant les caractères sur 8bits. Mais plusieurs de ces normes étaient nécessaire du fait de l'impossibilité de faire tenir les caractères de toutes les langues du monde sur 8bits. La norme iso-8859-1 latin1, la plus utilisée, se destine par exemple au langues d'europe de l'ouest, du nord, d'amérique, la norme iso-8859-2 latin2 est prévue pour l'europe de l'est...

L'unicode offre l'avantage de proposer les caractères nécessaires à toutes les langues du monde, mais reste encore peu utilisé ou implémenté.

En effet, il est impossible de l'utiliser en pratique sous linux-console, la mémoire exigüe de la carte vidéo ne permettant pas de stocker tous les caractères de l'unicode, et sous X la gestion actuelle des polices rendrait son utilisation trop inconfortable.

Attendos un peu que les polices «True Type» se répendent sous X, grace a des outils comme freetype, xfstt (...).

Actuellement, la nouvelle norme européenne iso-8859-15 latin9 souvent abbrégée en latin0 est censée permettre la transition vers l'unicode en apportant à l'ancien iso-8859-1 latin1 les caractères qu'il manquait actuellement.

Cet HOWTO sera mis à jour sous peu pour vous permettre de transiter vers le latin0 en toute sérénité, pour utiliser l'EURO par exemple.

3.8 UTC, GMT

Nous n'utilise pas les mêmes normes horaires que les ango-américains !

La France se trouve en zone MET (heure moyenne européenne) GMT+1 ; la zone «MET DST» correspond à l'heure d'été activée GMT+2.

Voici donc une petite explication de la différence entre GMT et UTC par Edgar Bonet Orozco :

GMT est l'heure solaire moyenne de Greenwich. L'heure est comptée à partir de midi et est donc décalée de 12 h par rapport au temps universel. GMT est basée sur la rotation de la Terre, ce n'est donc pas un temps régulier. En effet, la vitesse de rotation de la Terre subit des variations faibles mais assez cahotiques, en plus elle diminue sur le long terme.

UTC (temps universel coordonné) est compté à partir de minuit (toujours par rapport au méridien zéro). Il est basé sur des étalons atomiques, mais des secondes intercalaires sont ajoutées occasionnellement pour faire en sorte qu'il ne s'écarte jamais de plus de une seconde de l'heure solaire moyenne. Tu auras plus de précisions sur la détermination de UTC dans http ://opdaf1.obspm.fr/, et en particulier dans http ://opdaf1.obspm.fr/www/lexique.html.

C'est le temps UTC qui sert de référence aux différentes heures légales. L'acronyme «GMT» est employé 99% du temps pour désigner UTC, mais c'est une erreur (ou un abus de langage).

Enfin, il existe l'heure astronomique, aussi comptée sur des étalons atomiques mais décalée de 4 minutes par jour de l'heure terrestre car se basant sur la position de la terre dans l'espace par rapport aux étoiles.

4 Réglage du clavier sous Linux

4.1 Introduction

Il existe deux programmes pour configurer votre clavier : un pour la console : `loadkeys` et un pour XFree : `xkbd`.

Pour installer un clavier français sous Linux, tapez selon que vous soyez respectivement Français, Canadien Français, Belge Wallon ou Suisse Romand :

```
loadkeys french.map
loadkeys cf.map
loadkeys be-latin1.map
loadkeys sf-latin1.map
```

4.2 Où trouver ces fichiers

`french.map` est inclu dans ce HOWTO (au chapitre «Fichiers inclus») et tous les autres fichiers se trouvent d'ordinaire dans le répertoire `/usr/lib/kbd/keytables` ou dans `/usr/share/keytables` avec les distributions debian.

Pour l'installer, vous pouvez donc le recopier ou plus simplement le télécharger par ftp.

Tapez donc :

```
ftp ftp.linux.france.com
cd pub/
cd french
get french.map
exit
```

```
su root -c "cp french.map /usr/lib/kbd/keytables/"
```

4.3 Attention au stty !

Avec french.map et french (pour X window : cf. chapitre «Régler le clavier pour X window»), `N'UTILISEZ PLUS` :

```
stty erase ^H
```

ou

```
stty erase ^?
```

«stty erase» ne sert en effet qu'à établir une correspondance entre le code que renvoie une touche et la fonction `erase` (effacer le caractère).

Les tables ici fournies fonctionnent correctement et ne nécessitent donc pas de `stty erase` qui risquerait surtout de perturber leur bon fonctionnement !

Supprimez-donc ces lignes de vos fichier d'initialisation de l'interpréteur de commandes («shell»), comme .bashrc, .bash_profile ou .tcshrc .

4.4 Faire de ces modifications les défaut au démarrage

D'habitude, loadkeys est exécuté par des scripts au démarrage; par exemple dans `/etc/rc.d/init.d/keytable` ou `/etc/rc.d/rc.local`, `init.d/keytable` ou `rc.keymap`.

La commande :

```
find /etc/rc.d -type f | xargs grep loadkeys
```

vous fournira sur le nom exact du fichier lançant loadkeys; il vous suffit alors de l'éditer pour que vos modifications soient désormais prise en en compte à chaque démarrage.

Les distributions RedHat constituent une exception : la table de clavier par défaut étant définie dans `/etc/sysconfig/keyboard`.

Remplacer donc dans le fichier propre à votre distribution la table de clavier en question par `french.map`.

4.5 Vieilles versions

Si vous rencontrez les messages d'erreur suivants :

```
(guylhem@barberouge:guylhem)$ loadkeys french
Loading /usr/lib/kbd/keytables/french.map
unknown keysym 'dead_cedilla'
/usr/lib/kbd/keytables/french.map:67: parse error
syntax error in map file
key bindings not changed
(guylhem@barberouge:guylhem)$ loadkeys french
```

Celà signifie que votre version de kbd (dont le programme loadkeys dépend) est trop vieille : il faut vous procurer une version plus récente du programme.

Il existait et existe encore des versions des tables de clavier pour ceux ne voulant ni touches mortes ni accents, mais celles-ci tombent actuellement en désuétude puisque la grande majorité des programmes prennent désormais en charge ces fameuses touches mortes et les caractères accentués.

De plus, les tables fournies avec ce HOWTO permettent de se servir des touches mortes ou de s'en passer si un programme les refuse, les deux cas ayant été prévus.

5 Réglage du clavier sous XFree

5.1 Introduction

«xmodmap» servait à changer les tables de clavier sous XFree, un fichier de configuration d'ordinaire nommé /usr/X11/lib/X11/xinit/.Xmodmap étant pris en charge grâce à votre `.xinitrc` (à lire!).

Pour convertir une table console correcte, on utilisait «mk_modmap» (livré avec l'archive 'kbd' et généralement installé en /usr/lib/kbd/keytables)

Les utilisateurs désirant personnaliser leur table xmodmap employaient le programme «xkeycaps».

XFree 2.1 et les version ultérieures reconnaissent la disposition des touches du clavier gérée par «loadkeys», cela rendant théoriquement xmodmap inutile.

Mais la gestion de la touche «COMPOSE» assurée par XFree 3.1 laissait un peu à désirer, la table héritée de «loadkeys» ne servant guère.

Il fallait donc continuer à employer xmodmap, l'utilitaire logiciel standard permettant d'obliger le serveur X à associer les événements souris/clavier à des caracères.

Sous XFree version 3.2 et postérieures, «xkb» rendit xmodmap définitivement obsolete.

Mais ne voilà-t-il pas les rumeurs qui voulaient que les prochaines versions de XFree (3.3.1) ne supporte plus les touches mortes se sont confirmées!!!

Dans la distribution X11 du X Consortium (donc XFree86, qui en est directement issu), il n'existe plus de mécanisme général de gestion des touches mortes, ou des solutions approximatives obligeant à supprimer (aussi) xkb!

Quelle histoire compliquée!

De ce fait, elles ne fonctionnents que pour les clients X «internationalisés», c'est-à-dire qui gèrent eux-mêmes les problèmes liés aux méthodes d'entrée de données des différentes langues nationales.

Ceci est très gênant lorsqu'on veut pouvoir saisir du texte comportant des lettres accentuées, alors que le client utilisé n'est pas internationalisé!

Thomas Quinot (Thomas.Quinot à cuivre.fdn.fr) vous propose donc sa `libX11` modifiée, qui gère les touches mortes de manière transparente pour tous les clients.

Elle permet également de traiter les séquences composées inconnues comme des paires de caractères, par example :

`~ + / = ~/`

Pour l'utiliser :

- Récupérez libX11-XF3.3.1.tar.gz ;
- Extrayez le fichier `libX11.so.6.1` ;
- Placez-le dans `/usr/X11R6/lib/` ;
- Exécutez `ldconfig` ;
- Désactivez l'extension XKB en utilisant l'option `XkbDisable` dans la section `Keyboard` de votre `XF86Config`.

5.2 Conseils

Personellement, je vous recommande de ne pas toucher à la libX11 mais d'utiliser les alternatives du fichier french en attendant un autre changement de la politique du X11 Consortium envers les touches mortes ou en rajoutant les options d'internationalisation au code source de vos programmes.

Actuellement, si vous n'avez pas de répertoire `/usr/X11R6/lib/X11/xkb/symbols/` , procurez-vous donc une version plus récente de X window !

5.3 Est-ce nécessaire ?

Sans xkb, XFree se rabattra par défaut sur votre configuration de clavier pour Linux (avec loadkeys). Toutefois, la translation n'est pas parfaite et je vous recommanderais plutôt d'éviter cette option, aussi bien que l'ancien gestionnaire de clavier `Xmodmap`.

Il se peut aussi que sans aucune commande pour xkb, XFree vous considère comme utilisateur d'un clavier anglo-américain !

Il convient donc d'utiliser «fr» ou «french» avec xkb ; toutefois ce dernier permet d'utiliser de multiples améliorations pour disposer du plus de confort possible (gage de confort : je l'utilise actuellement :-)).

5.4 «french» ou «fr» ?

Je vous conseille de choisir `french` que je maintient à jour, plus récent que `fr` et plus adapté car maintenu par un francophone :-)

De plus, il bénéficie de nombreuses améliorations pour pallier au déficiences de nombreux programmes.

Par exemple, certains programmes en Motif comme Netscape (tm) ne savent pas encore gérer les touches mortes ... donc impossible de taper par exemple des ê ou des î sur les claviers belges et français, puisque la touche morte «accent circonflexe» est le seul moyen d'obtenir ces caractères !

«french» répond à ce problème en proposant une solution de remplacement : AltGr (voyelle) permet d'obtenir la voyelle avec un accent grave, AltGr+Shift (voyelle) permet d'obtenir la voyelle avec un tréma !

5.5 Table de correspondances

- AltGr (voyelle) : voyelle accent grave
- AltGr + Shift (voyelle) : voyelle tréma
- AltGr (touche au dessous de la voyelle) : voyelle majuscule accent grave
- AltGr + Shift (touche au dessous de la voyelle) : voyelle majuscule tréma
- AltGr + Shift (minuscule accentuée) : majuscule accentuée

Par exemple :

- AltGr (e) : ê
- AltGr + Shift (ù) : Ù

L'astuce marche aussi pour le c cédille :

- AltGr + Shift (ç) : Ç

Donc même dans les pires conditions (si aucune touche morte ne fonctionne), on peut quand même accéder à toutes les voyelles accentuées, majuscules et minuscules !

Vous y trouverez en plus les guillemets français (« »), les symboles employés en physique (ß, å, delta, phi ...), des signes de monnaies (yen, cent) ou de copyright (^{TM}, ©) ainsi que des lettres d'autres langues (Ø, Æ ...).

Tous ces signes s'obtiennent grâce à AltGr (lettre), par exemple AltGr (b) donne ß, AltGr (r) donne ^{TM}, AltGr (y) donne yen...

Pour ce qui est du choix entre les deux... à vous de voir !

Un conseil : essayez l'un puis l'autre.

5.6 Faire de ces modification les défaut au lancement de X

Pour XFree, votre fichier `XF86Config` doit contenir une section Keyboard :

```
Section "Keyboard"
Protocol        "Standard"
AutoRepeat      500 30
LeftAlt         Meta
RightAlt        ModeShift               # Important pour Alt-gr, mais dans les
                                        # versions récentes de XFree,
                                        # AltGr et RightAlt sont synonymes.
ScrollLock      Compose                 # Pour faire des caractères spéciaux.
                                        # Utiliser sinon une des 3 touches des
                                        # nouveaux claviers 105 touches...

RightCtl        Control                 # Garder la touche normale


XkbKeycodes     "xfree86"       #
XkbTypes        "default"       #
XkbCompat       "default"       #
XkbSymbols      "french(pc102)" #
XkbGeometry     "pc(pc102)"     #  Tout est nécessaire pour xkb
XkbModel        "pc102"         #
XkbLayout       "french"        #
XkbRules        "xfree86"       #
# Remplacer «french» par «ca» ou «be» ou «fr_CH»
EndSection
```

Pour cette partie Xkb, certains préfèrent :

```
XkbKeymap "xfree86(french)"
XkbKeycodes  "xfree86"
XkbSymbols   "en_US(pc101)+french"
XkbGeometry  "pc(pc101)"
```

5.7 Où trouver ces fichiers

`french` est inclu dans ce HOWTO (au chapitre «Fichiers inclus») et tous les autres fichiers se trouvent d'ordinaire dans le répertoire `/usr/X11R6/lib/X11/xkb/symbols/`.

Pour l'installer, vous pouvez donc le recopier ou plus simplement le télécharger par ftp.

Tapez donc :

```
ftp ftp.linux.france.com
cd pub/
cd french
get french
exit
su root -c "cp french /usr/X11R6/lib/X11/xkb/symbols/"
```

Mettre «french» dans `/usr/X11R6/lib/X11/xkb/symbols/french` et « french.map» dans `/usr/lib/kbd/keytables/french.map` ou `/usr/share/keytables/french.map`.

5.8 Autre méthodes

Une autre méthode pour obtenir des caratères 8 bits :

Dans `/usr/X11R6/lib/X11/locale/iso8859-1/Compose` se trouve une table de «correspondance» entre deux caractères et un résultat ; les deux caractères sont «mélangés» lors de l'appui sur la touche Compose : par exemple ;

```
<Touche COMPOSE> e ^
```

a toutes les chances de vous donner ê

Pour utiliser la touche Compose, vous devez l'affecter à une touche !

Le paramétrage préconisé, réglé dans XF86Config, la fait correspondre à la touche «Arrêt Défil» par la commande :

`ScrollLock Compose`

dans la section keyboard.

Éditez sinon votre table de clavier et remplacez par exemple :

```
key <TLDE> {    [       twosuperior,    threesuperior   ]       };
```

par :

```
key <TLDE> {            [       twosuperior,    threesuperior   ],
                        [       Compose                         ]       };
```

Si vous avez par exemple un clavier belge ou français, désormais appuyer sur la touche (carré/cube) vous donnera normalement (carré) en minuscules, (cube) en majuscules et en AltGr la fameuse fonction Compose ; vous pouvez bien sur décider d'attribuer Compose à une autre touche ; je vous recommande toutefois d'utiliser ScrollLock, bien moins utilisée que les autres touches.

NB : La touche carré/cube), peu utilisée, est par défaut assignée aux guillemets français carré et cube restant accessible respectivement en AltGr et AltGr + Shift ainsi que par l'accent circonflexe suivi du chiffre.

Pour modifier ce comportement par défaut, il vous suffit avant d'utiliser les fichiers de configuration pour XFree et linux en mode console de repérer la ligne mise en commentaire au début du fichier et de supprimer ce commentaire !

6 Réglage du clavier pour les applications en mode texte

6.1 Tout d'abord

Le huitième bit doit survivre à l'entrée du noyau, assurez-vous-en donc avec :

```
stty cs8 -istrip -parenb
```

Ensuite, il convient de s'assurer que l'application est apte à supporter les caractères 8 bits : par exemple, ispell n'est pas compilé pour des caractères 8 bits par défaut : il faut le recompiler sans l'option :

`-NO8BIT`

6.2 Appeler la police

Charger une police de caractères adaptées :

`setfont /usr/lib/kbd/consolefonts/xxx-NNN`

- xxx repésente le type de police ; il doit s'agir d'une iso01/lat1 ou iso15/lat0 pour les francophones sinon il n'y a plus de caractères «étendus» (voyelles accentués, symboles de ponctuation comme paragraphe...)
- NNN représente la taille individuelle de chaque caractère ; 08 est presque illisible mais 16 est un peu gros... À régler selon les goûts de chacun.

6.3 Une police, des polices...

De nous jours, une nouvelle norme latin0 (iso-8859-15) vient d'apparaître, destinée à remplacer l'ensemble des polices européennes par une seule.

Son but est voisin de l'unicode, mais elle à l'avantage d'être plus légère que celui-ci et d'être applicable tout de suite.

Elle apporte notamment des caractères qui manquaient beaucoup pour les français et les finlandais :

- «e dans l'o» en majuscule et minuscule pour les français, par exemple pour les mots oeuf, coeur, boeuf, oeil ...
- «s chapeau inversé» en majuscule et minuscule pour les finlandais
- «z chapeau inversé» en majuscule et minuscule pour les finlandais
- «l'EURO» pour tous les pays européens

Cette nouvelle norme, iso-8859-15 latin9 aussi connue sous le nom générique latin0, reste cependant encore très peu répendue par rapport à la norme actuelle iso-8859-1 latin1.

Pour l'utiliser, vous devez télécharger les fichiers correspondants sur par ftp.

Tapez donc :

```
ftp ftp.linux.france.com
cd pub/
cd french
get latin0.tgz
exit
```

Vous trouverez dans l'archive latin0.tgz :

- des polices .psf à mettre dans `/usr/lib/kbd/consolefonts/`, que l'on peut essayer avec setfont
- des polices .pcf à mettre dans `/usr/X11R6/lib/X11/fonts/misc/`, que l'on peut utiliser en tapant sous root :

```
cd /usr/X11R6/lib/X11/fonts/misc/
mkfontdir
xset +fp /usr/X11R6/lib/X11/fonts/misc/
```

Puis en choissant dans xfontsel les polices d'«encdng» (encodage) 15 à utiliser avec ses diverses applications.

- de nouvelles tables de clavier pour linux-console et X supportant les nouveaux caractères latin9

En attendant, Éric Dubreuil (ericdub à club-internet.fr) nous conseille la police `lat1u-16.psf`, permettant d'avoir tous les caractères accentuées sans pour autant enlaidir `mc`, un gestionnaire de fichier très populaire mais très sensible à la police utilisée !

Une alternative est l'unicode, mais les programmes l'utilisant sont encore trop peu nombreux pour qu'elle soit valable.

Citons quand même yuedit et le projet «9» (9term, 9wm...) qui utilisent l'unicode sous X, avec plus ou moins de succès.

En revanche, sous linux-console, tout programme peut utiliser l'unicode.

Mais, hélas !, aussi grande que soit la mémoire des cartes vidéos actuelles, elle ne peut charger l'intégralité des caractères d'une police unicode.

Il existe donc des «jeux de caractères» limités mais bien utiles pour les langues baltiques, asiatiques, est-africaines...

Essayez par exemple ce fichier `unicode-lance` pour passer en mode unicode :

```
#!/bin/sh
echo -e '\033%8'
kbd_mode -u
loadkeys french-unicode.map
setfont lat1u-16.psf -m null
```

et ce fichier `latin-retourne` pour revenir en mode latin normal :

```
#!/bin/sh
echo -e '\033%@'
kbd_mode -a
loadkeys french.map
setfont lat1u-16.psf -m null
```

6.4 Les messages en français

Linux en 'version française' existe... mais ne semble pas facile à obtenir.

La librairie locale(7) utilisée par la librairie C se sert de variables d'environnement pour définir les préférences linguistiques et nationales

- LC_COLLATE équivalence de caractères pour les comparaisons. (ex : æ peut être équivalent à ae)
- LC_CTYPE caractères affichables
- LC_MONETARY monnaie utilisée
- LC_MESSAGES langue des messages
- LC_TIME format de la date
- LC_ALL valeur par défaut des variables précedentes
- LANG surpasse les variables précédentes.

Les valeurs utilisables pour le français sont :

- fr français générique

- fr_FR français de France
- fr_BE français de Belgique
- fr_CH français de Suisse
- fr_LU français du Luxembourg
- fr_CA français du Canada

Par exemple, si la monnaie vous importe peu, vous pouvez indiquer

```
export LANG=fr
```

Les données correspondantes aux préférences se trouvent généralement dans `/usr/share/locale`. S'ils manquent, vous pouvez récuperer leur source sur `ftp ://dkuug.dk/i18n/WG15-collection`

Pour se faire `SON` linux en français il faut en pratique disposer des sources de la libc et d'un minimum de compétences (langage C).

Voici donc, pour la bonne bouche, quelques renseignements collectés...

Laurent Chemla nous pilote : « Si j'ai suivi, il y a deux trucs qui font la meme chose, d'ou embrouille, les locales et un truc appelé nls.

Les locales, sous Linux, ça va par défaut dans /usr/lib/locale, et ça contient de quoi préciser entre autres le format des nombres (LC_NUMERIC), les retours des fonctions ctype (isalpha etc) (LC_CTYPE), le format des sommes d'argent (LC_MONETARY), le format des dates (LC_TIME) et encore d'autres détails. Sur certains Unix, il y a aussi LC_MESSAGES, qui contient les messages de l'OS dans la langue choisie par la variable d'environnement LC_MESSAGES. Pas sous Linux. Sous Linux, c'est nls qui fait ça, et de fait, nls gêre un fichier qui est dans /etc/locale/. (NDG : De nos jours, ils se trouvent plutôt dans /usr/lib/locale) D'où l'embrouille. Ils auraient mis ça dans /usr/lib/nls, comme tout le monde, ça aurait evite la confusion. Mais bon.

Donc, si l'on veut se tenir à jour d'nls, il suffit d'aller dans les sources de libc/nls, faire 'make french' et copier libc.dat dans /etc/locale/C/libc.dat et toc, tous les messages sont en francais. Y compris ceux des programmes qui n'utilisent pas setlocale() : c'est perror() qui fait le boulot (du moins je suppose).

Si l'on veut en plus que les programmes qui utilisent setlocale() utilisent nos règles françaises, là il faut aller dans les sources de libc/locale et dans chacun des sous-répertoires jeter un oeil sur le fichier exemple, le modifier pour le français (c'est des tout petits fichiers), et faire 'make', avant de copier le résultat dans /usr/lib/local/fr/ et de faire un 'export LC_ALL=fr'. »

Autre astuce pour ménager les vieux logiciels :

```
cd /usr/lib ln -s /usr/share/locale
```

Nat Makarévitch rapelle à juste titre que site FTP du LIP6 offre de nombreux documents et fichiers utiles, en particulier :

- Les pages de manuel en français (archive man-fr-*) adaptées par C. Blaess. Il suffit d'extraire le contenu de l'archive dans le répertoire /usr/man/fr.
- Une intéressante documentation technique : locale.fr (P. D'Cruze, adaptée par É. Dumas)

Pour en apprendre davantage lire la page de manuel de «locale» (bibliothèque de pages de manuel 5 ou 7).

6.5 Les variables

Ces variables doivent être exportées à chaque login : mettez les donc dans `.bashrc` et `.bash_profile`, `.tcshrc` ou /etc/profile :

```
# pour tcsh, utiliser setenv au lieu de export, par exemple «setenv LANG fr»

# Les locales
LC_CTYPE=ISO-8859-1
LANGUAGE=fr
LC_MESSAGES=fr
LC_ALL=fr
LANG=fr
LESSCHARSET=latin1
export LC_CTYPE LANGUAGE LC_MESSAGES LC_ALL LANG LESSCHARSET

# Les raccourcis : sans ll ou d, impossible de voir les fichiers comprenant des
# caractères 8 bits ...
alias q="cd .."
alias ll="ls --color=auto -a -N -l"
alias d="ls --color=auto -a -N"
alias indent="indent -kr"
alias netscape="export LC_ALL=en_EN ; netscape $* ; LC_ALL=fr_FR"

# Au cas où del/backspace ne fonctionne pas, essayez l'un des deux :
#stty erase ^?
#stty erase ^H
```

N'oubliez pas : tous les fichiers ici cités dont le nom commence par «.» doivent se trouver dans votre répertoire maison !

6.6 Les caractères 8 bits dans les programmes utilisant readline tels bash ou ncftp...

Readline est une librairie de saisie de caractères avec historique et completion; elle est utilisée par de nombreux programmes (bash, ncftp, gnuplot...).

Par défaut, le 8 ème bit sert à coder la touche Meta.

Elle est utilisée pour coder certaines commandes mais la touche Escape (ou Echap) peut la remplacer.

Si vous souhaitez voir et saisir des accents avec les applications compliées avec readline, comme bash pour les versions supérieures à 1.13, ajoutez ceci dans votre fichier $INPUTRC (par défaut `~/.inputrc`) :

```
# Permettre de rentrer & recevoir des caractères accentués
set meta-flag on
set convert-meta off
set input-meta on
set output-meta on

# Pas de bip audible mais visible
# set bell-style visible

# Pour faire marcher Home, End, Delete, Esc + Delete & Esc + Backspace
"\e[1":beginning-of-line
"\e[3":delete-char
"\e[4":end-of-line
"\e\C-h": backward-kill-word
"\e\e[3": kill-word
```

Celà vous permettra d'effacer des caractères vers l'avant avec Delete et vers l'arrière avec Backspace, ceci

n'étant pas prévu sur les tables de clavier par défaut «mimant» le comportement d'un clavier de console VT100 .

Les deux dernières options vous permettront d'utiliser Home et End pour aller au début et à la fin de la ligne en cours d'édition.

Pour pouvoir employer une touche il suffit d'obtenir son code et l'associer à une fonction grâce à ce fichier ; pour obtenir le code d'une touche invoquer cat puis appuyer sur la touche : son code apparaît.

Le code physique :

```
^[
```

correspond à la séquence d'échappement.

Par exemple, si le code de «Home» est ^[[7 il suffit d'ajouter une ligne

```
"\e[7~": beginning-of-line
```

au `.inputrc`.

Plus de documentation est disponible sur la page de manuel de bash, section `READLINE`.

On peut aussi, grâce à la variable d'environnement `INPUTRC`, utiliser un fichier commun à tous les utilisateurs.

Enfin, je vous recommande d'utiliser ce fichier `.bashrc` (à placer dans votre répertoire maison) :

```
# ~/.bashrc : executé par bash(1) pour les shells lancés après le login.

# Le path
PATH="/usr/sbin:/usr/bin:/sbin:/bin:/usr/X11R6/bin:/usr/local/bin"

# Le prompt
PS1="(\u@\h:\W)$ "

# L'utilisateur
ENV=$HOME/.bashrc
USERNAME="votre nom"

# Les locales
_CTYPE=ISO-8859-1
LANGUAGE=fr
LC_MESSAGES=fr
LC_ALL=fr
LANG=fr
LESSCHARSET=latin1
export LC_CTYPE LANGUAGE LC_MESSAGES LC_ALL LANG LESSCHARSET

# Les raccourcis
alias q="cd .."
alias ll="ls --color=auto -a -N -l"
alias d="ls --color=auto -a -N"
alias indent="indent -kr"
alias elm-box="elm -f ~/Mail/Inbox"
alias ftp=ncftp
alias tin=rtin
alias netscape="export LC_ALL=en_EN ; netscape $* ; LC_ALL=fr_FR"

# Au cas où del/backspace ne fonctionne pas, essayez l'un des deux :
```

```
#stty erase
#stty erase ^H

# Refuser le talk & le write
#mesg n

# Permettre les coredumps
#ulimit -c nombre_max_de_Ko

#Fixer le masque de creation de fichiers
#umask 022
```

Le fichier .bash_profile executé par bash(1) lors les logins, y est strictement identique mais peut, si vous le voulez, présenter des différences pour afficher par exemple une citation en laçant le programme «fortune».

6.7 zsh

Celui-ci se contente d'un `stty pass8`.

6.8 tcsh

Pour tcsh, placer, dans le fichier `/etc/csh.login` (ou dans les fichiers `.tcshrc` des utilisateurs concernés) les variables indiquées plus haut en remplaçant export par setenv :

```
setenv LC_CTYPE=ISO-8859-1
setenv LANGUAGE=fr
setenv LC_MESSAGES=fr
setenv LC_ALL=fr
setenv LANG=fr
setenv LESSCHARSET=latin1
```

6.9 Pour tous les shells

Remplacer 'iso_8859_1' par 'iso-8859-1' si cela ne fonctionne pas, voire french.iso88591.fr_FR (ou fr_BR, fr_CA, fr_CH, fr_LU ...)

À propos de LANG, Pablo Saratxaga signale qu'il vaut mieux éviter fr_FR : «Par ailleurs je conseille un simple "fr" plutôt que fr_FR etc. Certains programmes n'aiment pas trop (ils ne passent pas vraiment par la resolution de locales mais lisent LANG directemment...)»

Si nls est installé, les routines correspondantes sont utilisées, sinon tcsh agit en iso_8859_1, quelle que soit les valeurs données à LANG et LC_CTYPE : cf. la section NATIVE LANGUAGE SYSTEM de tcsh(1).

Selon le Danish-HOWTO, utiliser :

```
(guylhem@barberouge:tcsh)# setenv LC_CTYPE ISO-8859-1
(guylhem@barberouge:tcsh)# stty pass8
```

6.10 L'heure

Rien n'est plus facile que de laisser à Linux le soin de passer de l'heure d'été à l'heure d'hiver et réciproquement...

a) D'abord, quelle heure est-il :-) ?

Après avoir regardé sur la pendule la plus proche de vous, tapez :

```
clock
```

puis

```
date
```

La première heure est celle du bios, la seconde celle calculée par Linux à partir de `/etc/zoneconfig`.

Pour peu que vous ayez installé timezone, votre machine peut jongler aisément entre les différents fuseaux horaires... ce que les DOS ou Windows ne permettent pas !

b) L'heure du choix !

De là, deux possibilitées : soit vous décidez de laisser l'heure système à l'heure locale (peu pratique, excepté si vous hébergez aussi sur votre disque dur ces systèmes d'exploitation peu évolué, ne sachant même pas exploiter les fuseaux horaires), soit vous passez à l'heure de Greenwich, Linux se chargeant de l'adapter à votre fuseau horaire.

Dans ce cas, les grands voyages à l'étranger avec un portable tout neuf équipé de Linux se feront en toute simplicité : le dernier pargraphe vous expliquera comment changer de fuseau horaire facilement...

c) Le réglage

Pour procéder, regardez d'abord si l'heure système retournée par clock est l'heure locale ou l'heure de Greenwich (vous avez bien regardé une pendule comme je vous le conseille plus haut, n'est-ce pas ?).

Si clock vous retourne l'heure locale, il va falloir jongler un peu avec les fuseaux horaires, sinon passez directement à l'étape «réglage du fuseau».

- Pour les français, belges wallons et suisses romands :

Vous êtes normalement dans une zone horaire MET (Medium European Time, soit GMT+1).

La zone «MET DST» correspond elle à l'heure d'été active (GMT + 2).

Selon les décisions futures de la cour européenne, l'heure d'été (daylight savings) risque d'être abandonné... ce qui compliquera cette explication !

- Pour les canadiens français :

Vous êtes dans une zone horaire GMT-4 si vous habitez autour de Montréal .

Pour les autres provinces, consulter un dictionnaire !

Premièrement, effacez le fichier /etc/localtime ou /usr/lib/zoneinfo/localtime (l'emplacement varie selon les distributions ; si vous avez les deux fichiers, effacez /etc/localtime : l'autre fichier doit certainement être un lien vers /etc/localtime).

Dès lors, clock et date doivent vous retourner la même heure...

- Pour les canadiens français : utilisez GMT+4

- Pour les français, belges wallons et suisses romands :

Si vous lisez ce document en été, le décalage de l'heure locale par rapport à l'heure de Greenwich est de deux heures, on va donc mettre temporairement le fuseau en GMT-2, soit vous lisez ce document en hiver et il convient de remplacer tous les «2» par des «1» dans les exemples suivants...

GMT-2 ? ! ! ? Alors que le fuseau est GMT+2 ?

Ceci va vous retourner l'heure de Greenwich dont on a besoin pour régler le système : tapez :

```
ln -sf /usr/lib/zoneinfo/Etc/GMT-2 /etc/zoneconfig
```

```
clock
```

```
date
```

L'heure calculée correspond donc à l'heure de Greenwich, l'heure système à l'heure locale... soit l'inverse de ce que l'on veut.

Un simple : `clock -w`

Va alors mettre l'heure du système en heure de Greenwich, utilisant pour celà l'heure locale inversée volontairement avec l'heure système.

d) Régler le fuseau horaire :

Tapez alors : `ln -sf /usr/lib/zoneinfo/Europe/Votre-ville /etc/zoneconfig`

Et voilà ! Le système est à l'heure de Greenwich, la conversion vers le fuseau horaire local et l'heure d'été se faisant automatiquement.

Si vous habitez hors Europe, procédez de même en remplaçant les «-2» par «X», X représentant l'opposé de votre décalage horaire (par exemple -4 au Québec, donc X=+4...)

e) Modifier le fuseau horaire :

Si vous partez à l'étranger, il est alors simple de modifier l'heure locale de votre portable : par exemple si vous partez pour Moscou :

```
ln -sf /usr/lib/zoneinfo/Europe/Moscow /etc/zoneconfig
```

Très simple et très utile : plus besoin de se souvenir du décalage horaire (plus ou moins X heures) puisque Linux se charge désormais de tout !

6.11 vi

Normalement, aucune modification n'est nécessaire à part l'utilisation de loadkyes (détaillée plus haut).

Ce fichier `.vimrc` pourrait toutefois vous être utile si vous utilisez `vim` au lieu d'`elvis` :

```
" Les guillemets anglais «"» signifient «mis en commentaire»

" Éviter un avertissement «version incorrecte» :
version 4.0

" Police à utiliser sous X11 :
"set guifont=-misc-fixed-medium-r-normal--14-130-75-75-c-70-iso8859-1

" Fixer la césure automatique de la ligne à N caractères :
set tw=79

" Indentation (pour les programmeurs) :
" set cindent

" Réglages souris :
"set mouse=a

" Ligne
set nowrapscan

" Montrer les correspondances :
```

```
set showmatch

" Montrer le mode
set showmode

" Indentation intelligente (pour les programmeurs) :
" set uc=0

" Faire fonctionner backspace :
set t_kD=^?

" Doit se trouver après
map ^H X

" Faire fonctionner delete  :
map \e[3~ x

" Cacher la souris lorsqu'on tape :
set mousehide

" Faire voir les correspondances lors de recherches :
" set hlsearch

" Colorer à l'intérieur des commentaires en C :
let c_comment_strings=1

" Couleur pour xiterm, rxvt, nxterm, color-xterm :
if has("terminfo")
    set t_Co=8
    set t_Sf=\e[3%p1%dm
    set t_Sb=\e[4%p1%dm
 else
    set t_Co=8
    set t_Sf=\e[3%dm
    set t_Sb=\e[4%dm
endif

" Coloration syntaxique :
if ett_Co > 1
   syntax on
endif

" Autoriser l'édition de fichiers gzippés

augroup gzip
  " Supprimer toutes les autocommandes
  au!
  autocmd BufReadPre,FileReadPre        *.gz set bin
  autocmd BufReadPost,FileReadPost      *.gz '[,']!gunzip
  autocmd BufReadPost,FileReadPost      *.gz set nobin
  autocmd BufReadPost,FileReadPost      *.gz execute ":doautocmd BufReadPost " .  expand("%:r")

  autocmd BufWritePost,FileWritePost    *.gz !mv <afile> <afile>:r
  autocmd BufWritePost,FileWritePost    *.gz !gzip <afile>:r
```

```
  autocmd FileAppendPre                *.gz !gunzip <afile>
  autocmd FileAppendPre                *.gz !mv <afile>:r <afile>
  autocmd FileAppendPost               *.gz !mv <afile> <afile>:r
  autocmd FileAppendPost               *.gz !gzip <afile>:r
augroup END
```

6.12 Emacs

Voici un fichier de configuration de base pour employer les accents. XEmacs les accepte lui par défaut sans broncher :-)

```
;;
;; Fichier .emacs: initialisation d'emacs
;; Tiré du Guide du Rootard
;;

(display-time)                      ;; Pour avoir l'heure dans la barre d'état
(setq display-time-24hr-format t)  ;; Format 24 heures

;; Nouveaux modes
(autoload 'c++-mode   "cplus-md" "C++ Editing Mode" t)
(autoload 'perl-mode   "perl-mode" "Perl Editing Mode" t)
(autoload 'c-mode "c-mode" "C Editing Mode" t)
; mieux vaudrait utiliser le "cc-mode"

(autoload 'jargon-mode "jargon" "Jargon Mode" t)

;; Auto-Mode Settings : positionne le mode selon l'extension
(setq auto-mode-alist
(append '(("\.c$"  . c-mode)    ;; utilise le mode C++ même pour C
("\.h$"  . c-mode)
("\.C$"  . c++-mode)
("\.H$"  . c++-mode)
("\.cc$" . c++-mode)
("\.C$"  . c++-mode)
("\.pl$" . perl-mode)           ;; Perl
("/tmp/snd\.[0-9]*" . text-mode);; Text (pour le mail)
("[Rr][Ee][0-9]*" . text-mode)
("\.ada$"  . ada-mode)          ;; Ada
("\.spec$"  . ada-mode)
("\.body$"  . ada-mode)
("makefile$" . makefile-mode)  ;; Makefile
("Makefile$" . makefile-mode)
("Imakefile$" . makefile-mode))
auto-mode-alist))

; Remapes varies a mettre ici
(global-set-key "\eg" 'goto-line)       ;; ESC G = Aller à une ligne

(put 'eval-expression 'disabled nil)

;; Accents...
(standard-display-european 1)
(load-library "iso-syntax")
```

```
(set-input-mode (car (current-input-mode))
                (nth 1 (current-input-mode))
                0)

;; Sous X-Window, texte en couleurs (C/C++/Shell/Makefile,etc)
(cond (window-system
(setq hilit-mode-enable-list  '(not text-mode)
hilit-background-mode   'light
hilit-inhibit-hooks     nil
hilit-inhibit-rebinding nil)
(require 'hilit19)
))
(if (not (equal window-system ""))
(global-set-key "\C-?" 'delete-char))
))
```

Le Keyboard-HOWTO conseille quand à lui :

```
(standard-display-european t)
(set-input-mode nil nil 1)
(require 'iso-syntax)
(load-library "iso-insert.el")
(define-key global-map [?\C-.] 8859-1-map)
```

Nat Makarévitch suggère d'utiliser une version au moins égale à 19.27, modifier le fichier global (commun à tous les utilisateurs) `/usr/lib/emacs/site-lisp/site-start.el` ou le `./emacs` de chaque utilisateur concerné pour y ajouter :

```
(standard-display-european t)
(set-input-mode (car (current-input-mode))
  (nth 1 (current-input-mode))
  0)
(global-set-key [delete]   'delete-char)
(global-set-key [home]     'beginning-of-line)
(global-set-key [end]      'end-of-line)
(global-set-key [prior]    'scroll-down)
(global-set-key [next]     'scroll-up)
(global-set-key [C-right]  'forward-word)
(global-set-key [C-left]   'backward-word)
(global-set-key [C-prior]  'beginning-of-buffer)
(global-set-key [C-next]   'end-of-buffer)
(global-set-key "\033[A"  'previous-line)
(global-set-key "\033[B"  'next-line)
(global-set-key "\033[C"  'forward-char)
(global-set-key "\033[D"  'backward-char)
(global-set-key "\033[H"  'beginning-of-line)
(global-set-key "\033[Y"  'end-of-line)
(global-set-key "\033[1~" 'beginning-of-line)
(global-set-key "\033[2~" 'overwrite-mode)
;; le delete-char peut correspondre à \004 ou à \033[3~
(global-set-key "\033[3~" 'delete-char)
(global-set-key "\033[4~" 'end-of-line)
(global-set-key "\033[5~" 'scroll-down)
(global-set-key "\033[6~" 'scroll-up)
```

Pour les autres versions :

– 19.19 :

```
(standard-display-european 1)
(set-input-mode (car (current-input-mode))
  (nth 1 (current-input-mode))
  0)
```

Lire à ce propos le fichier emacs.info.

– 19.22 :

```
(load-library "iso-transl")
(standard-display-european t)
```

– Autres :

```
(standard-display-european 1)
(load-library "iso-transl")
```

Le mode ispell d'emacs a un certain nombre de dictionaires prédéfinis qui ne sont pas forcement ceux qui sont en place sur le système.

Si vous constatez des problèmes, vous devez redéfinir la liste ispell-dictionary-alist des dictionnaires.

Cette liste contient :

– le nom du dictionnaire
– la liste des caractères composant un mot
– la lise opposée des caractères ne composant pas un mot
– les caractères de liaison à l'interieur des mots ("-", "'"...)
– vrai ou faux si ces caractères peuvent être présents en plusieurs exemplaire dans le mot
– une liste d'arguments pour ispell
– le mode d'ispell (tex, nroff..)

Par exemple, dans le fichier `.emacs` :

```
(setq ispell-dictionary-alist
 '((nil  ; francais.aff
  "[A-Za-zÀÂÇ-ËÎÏÔÙÛÜàâç-ëîïôùûü]" "[^A-Za-zÀÂÇ-ËÎÏÔÙÛÜàâç-ëîïôùûü]"
  "[---']" nil ("-n") "~nroff")
 ("english"  ; rosbif
  "[A-Za-z]" "[^A-Za-z]" "[---']" nil ("-B") nil)
 ("american" ; yankee
  "[A-Za-z]" "[^A-Za-z]" "[---']" nil nil nil)
 ("francais" ; français
  "[A-Za-zÀÂÇ-ËÎÏÔÙÛÜàâç-ëîïôùûü]" "[^A-Za-zÀÂÇ-ËÎÏÔÙÛÜàâç-ëîïôùûü]"
  "[---']" nil ("-n") "~nroff")
 ("francais-TeX" ; français
  "[A-Za-zÀÂÇ-ËÎÏÔÙÛÜàâç-ëîïôùûü\\]" "[^A-Za-zÀÂÇ-ËÎÏÔÙÛÜàâç-ëîïôùûü\\]"
  "[---'^`\"]" t nil "~tex")
 ("espanol" ; espa~nol.aff
  "[A-Za-záéíóúüñÁÉÍÓÚÜÑ]" "[^A-Za-záéíóúüñÁÉÍÓÚÜÑ]"
  "[---'^`\"]" t nil "~nroff")
 ("dansk" ; danois et norvégien
  "[A-Za-zåÅøØæÆéÉèÈ]" "[^A-Za-zåÅøØæÆéÉèÈ]"
  "[---'^`\"]" t nil "~nroff")
 ))
```

Le dictionnaire "francais-TeX" est un lien symbolique sur "francais" qui permet d'ajouter une entrée pour les accents à la TeX (ex : \'e).

Si vous utilisez X11, vous voudrez peut-être reconstruire les menus et il vous faudra recharger une partie de lisp/loaddefs.el dans votre `.emacs` :

```
(setq ispell-menu-map nil)

  (if ispell-menu-map-needed
  (let ((dicts (reverse (cons (cons "default" nil) ispell-dictionary-alist)))
    name)
    ;; Can put in defvar when external defines are removed.
    (setq ispell-menu-map (make-sparse-keymap "Spell"))
    (while dicts
  (setq name (car (car dicts))
    dicts (cdr dicts))
  (if (stringp name)
  (define-key ispell-menu-map (vector (intern name))
    (cons (concat "Select " (capitalize name))
  (list 'lambda () '(interactive)
    (list 'ispell-change-dictionary name))))))))

  (if ispell-menu-map-needed
  (progn
    ;; Define commands in opposite order you want them to appear in menu.
    (define-key ispell-menu-map [ispell-change-dictionary]
  '("Change Dictionary" . ispell-change-dictionary))
    (define-key ispell-menu-map [ispell-kill-ispell]
  '("Kill Process" . ispell-kill-ispell))
    (define-key ispell-menu-map [ispell-pdict-save]
  '("Save Dictionary" . (lambda () (interactive) (ispell-pdict-save t))))
    (define-key ispell-menu-map [ispell-complete-word]
  '("Complete Word" . ispell-complete-word))
    (define-key ispell-menu-map [ispell-complete-word-interior-frag]
  '("Complete Word Frag" . ispell-complete-word-interior-frag))
    (define-key ispell-menu-map [ispell-continue]
  '("Continue Check" . ispell-continue))
    (define-key ispell-menu-map [ispell-region]
  '("Check Region" . ispell-region))
    (define-key ispell-menu-map [ispell-word]
  '("Check Word" . ispell-word))
    (define-key ispell-menu-map [ispell-buffer]
  '("Check Buffer" . ispell-buffer))
    (define-key ispell-menu-map [ispell-message]
  '("Check Message" . ispell-message))
    (define-key ispell-menu-map [ispell-help]
  '("Help" . (lambda () (interactive) (describe-function 'ispell-help))))
    (put 'ispell-region 'menu-enable 'mark-active)))

  (fset 'ispell-menu-map (symbol-value 'ispell-menu-map))
```

6.13 less

`/etc/profile` ou les fichiers de démarrage du shell (comme `.bashrc` et `.bash_profile` dans votre répertoire maison) doivent contenir :

```
export LESSCHARSET=latin1
```

Comme fichier `.lessrc` je vous recommande :

```
\e[B    forw-line
\e[A    back-line
\e[6~   forw-scroll
\e[5~   back-scroll
\e[1~   goto-line
\e[4~   goto-end
\e[C    next-file
\e[D    prev-file
\eOA    back-line
\eOB    forw-line
```

Pour rentrer un Esc (parfois nommé échap) taper «\e» ou, sous vi, mettez vous en mode insertion avec i puis tapez ctrl+v et enfin esc.

Ce fichier permet d'utiliser les touches «Home», «End,», «PagerUp» & «PageDown».

Pour s'en servir, taper :

```
lesskey -o .less .lessrc
```

N'oubliez pas de mettre le fichier .less dans votre répertoire maison : il permet d'employer les touches fléchées, Page_up, Page_down, Home et End.

Ne pas négliger la page de manuel de less car il peut être intéressant de changer la valeur de sa variable d'environnement de configuration ("LESS").

J'utilise : '-C -M -i -x2'

6.14 ls

Utilisez les options :

```
-N --color=auto
```

ou, plus simplement, définissez dans les fichiers de démarrage de votre shell les alias suivants permettant de voir les fichiers dont le nom comprend des caractères 8 bits :

```
alias ll="ls --color=auto -N -l -a"
alias d="ls --color=auto -N"
```

Un simple `d` remplacera le classique `ls`, en y ajoutant la couleur, alors que `ll` vous offrira une liste complète de tous les fichiers présents dans le répertoire, avec leurs attributs, leurs tailles...

Pablo Saratxaga conseille d'utiliser –color=auto car «–color=yes fera avoir des codes escape ansi même lorsqu'on redirige vers un pipe un fichier où que le terminal ne supporte pas la couleur... ce n'est pas ce qu'on voudrait.»

6.15 flex

Donnez l'option :

```
-8
```

si l'analyseur généré doit accepter les entrees 8-bits (bien sur qu'il doit le faire!).

6.16 elm

Rajoutez ces trois lignes dans votre `.elm/elmrc` :

```
charset = iso-8859-1
displaycharset = iso-8859-1
textencoding = 8bit
```

6.17 col

Assurez-vous :

1) qu'il a été corrigé et fasse un

```
setlocale(LC_CTYPE,"");
```

2) de définir

```
LC_CTYPE=ISO-8859-1
```

dans l'environment.

6.18 rlogin

Utilisez l'option :

```
-8
```

6.19 joe

Utiliser `ftp ://sunsite.unc.edu :/pub/Linux/apps/editors/joe-2.8.tar.gz` ou plus récent qui devraient fonctionner après édition du fichier de configuration : placer l'option -asis dans `/usr/lib/joerc` en première colonne.

6.20 TeX et LaTeX

Le plus simple consiste à employer GuTemberg, une distribution de LaTeX avec tous les défauts réglés pour des francophones par des francophones !

Mais si vous préférez une version `standard` non modifiée, il suffit d'ajouter au début de chaque fichier :

- Pour LaTeX :

```
\documentstyle[isolatin]{article}
```

- Pour LaTeX2e :

```
\documentclass[a4paper,french]{letter}
\usepackage{isolatin1}
\usepackage[french]{babel}
\usepackage{t1enc}
```

  ou alors :

```
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
```

Au cas où votre distribution soit trop vieille pour l'inclure, `isolatin.sty` est disponible séparément sur : `ftp.vlsivie.tuwien.ac.at/pub/8bit`

Pour gs et xdvi, il faut utiliser des options speciales (sur ma machine, ce sont des alias). En effet, ils sont souvent configurés pour un papier de taille anglo-saxonne.

Ces options sont les suivantes :

```
gs -sPAPERSIZE=a4 xdvi -paper a4 ghostview -a4
```

Pour que dvips convertisse les documents dans un format papier a4, il faut specifier dans le fichier config.ps (le chemin varie en fonction des versions de LaTeX) : /usr/lib/texmf/dvips/config.ps ou `.dvipsrc`

```
@ a4 210mm 297mm @+! %%DocumentPaperSizes : a4 @+ %%PaperSize : a4 @+ %%BeginPaperSize :
a4 @+ a4
```

Sinon ajoutez à votre `.Xresources` :

```
XDvi.paper:             a4
Ghostview.pageMedia:    A4
```

6.21 Manuel

Tout d'abord, il faut vous procurer les pages de manuel Linux traduites en français sur `ftp.lip6.fr/pub/linux`

Si vous utilisez une distribution RedHat ou Debian, celles-ci sont présentes sur le cédérom : il suffit de les sélectionner lors de l'installation.

De là, deux possibilitées :

- soit vous ne voulez que des manuels en français, auquel cas il faut faire un :

  ```
  export MANPATH=/usr/man/fr_FR
  ```

  NB : fr_FR se remplace par fr_BE, fr_CA, fr_CH ou fr_LU selon le pays concerné.
- soit vous préférez par défaut des manuels en français et si la page n'existe pas, vous vous rabattez vers l'anglais, il suffir alors de taper :

  ```
  export LANG=fr
  ```

`man` ira chercher dans le «MANPATH» des pages en «LANG» par défaut et se rabattra sinon sur des pages en anglais.

6.22 Correcteur orthographique ispell

Vous pouvez vous le procurer sur son site original HYDRO-QUÉBEC : `ftp.ireq-robot.hydro.qc.ca :/pub/ispell/francais-IREQ-1.3.tar.gz`

ou sur : `ftp.lip6.fr/pub/ispell/francais`

Un fichier LISEZMOI explique pas à pas et en français l'installation.

6.23 Les claviers 105 touches

Aussi appelés «claviers Microsoft (tm)», ils ont trois touches de plus que les claviers normaux.

Comment s'en servir ?

En mode console, éditer `french.map` et décommenter les lignes en parlant !

De même pour `french` (pour Xwindow).

6.24 PGP

Tout d'abord, un avertissement : ce paragraphe n'est destiné qu'aux francophones ne vivant pas en France, même malgré les modifications récentes de la loi !

En effet, en France, en Iran, en Irak et dans quelques autres pays totalitaires, l'utilisation de logiciels de cryptographie est considérée comme usage d'armes de guerre, ce qui est punissable par la loi !

Donc sautez ce paragraphe si votre législation locale ne permet pas d'utiliser des logiciels de cryptographie.

Premièrement, il faut se procurer pgp-international (à cause de problèmes de patentes et de législation sur l'export d'armes de guerre, il existe pgp-us et pgp-international, ce dernire étant doté d'un algorythme encore plus puissant) sur : `ftp ://ftp.ifi.uio.no/pub/pgp/`

Ensuite, mettre dans votre répertoire maison, dans un sous répertoire .pgp les fichiers :

```
config.txt
language.idx
pgpdoc1.txt
pgpdoc2.txt
language.txt
fr.hlp
en.hlp
pgp.hlp
```

Enfin, éditer le fichier `.pgp/config.txt` pour y rajouter :

```
Language = fr
CharSet = latin1
```

Lisez ensuite `.pgp/fr.hlp` pour apprendre à vous servir de pgp !

6.25 Impression en mode ascii

Pierre-Charles David (pcdavid à club-internet.fr) vous donne ce truc :

Pour l'impression de texte simple, beaucoup d'imprimantes attendent des lettres en codage IBM (celui utilisé par MS-DOS).

Si le fichier est codé en latin1, on doit utiliser 'recode latin1 :ibmpc' (à placer dans les filtres lpr de /etc/ par exemple), sous peine de voir imprimer des 'alpha' à la place de 'à', et autres choses du même genre.

Il existe d'autres utilisation au programme `recode` : «ibmpc :lat1» ou «lat1 :ibmpc» (de ou vers MS-DOS) ou «applemac» pour Macintosh (tm).

6.26 unicode/latin/cp... je n'ai pas compris ce dernier paragraphe !

Voici donc plus d'explications grâce à Pablo Sartxaga (srtxg à chanae.alphanet.ch) dans un message explicant la différence unicode/latin/cp :

```
DB>   quand on lance "setfont" sans argument
DB>   il semble qu'on obtienne le meme effet que setfont -u none
```

Normal et logique.

DB> --> en un mot la table de conversion "unicode" est desactivee
DB> ce qui transforme le "e accent aigu (dec 130)" en "théta (dec 233)"

Non. Il ne transforme rien du tout, il affiche tout simplemment sans
aucune transformation au contraire, à toi de t'assurer que la fonte
que tu charges possèdes les glyphes à la bonne place par rapport au codage
de caractères que tu utilsies (iso-8859-1).

Autremment dit si ta fonte n'est pas iso-8859-1 tu dois dire à setfont de
faire des mappings; en particulier s ic'est une fonte qui utilise l'encodage
DOS cp437 tu fais "setfont -u cp437 nomdelafonte". Moi par exemple j'aime
beaucoup la fonte t.fnt de la Slackware, et j'ai donc quelque part
dans /etc/rc.d/* une ligne "setfont -u cp437 t.fnt"

Ou alors tu modifies la fonte (il y a un soft qui fait cela avec
interface svgalib, et un autre qui permet de créer des images an ascii-art,
un fichier par caractère (256 fichiers donc), il te suffit alors de jongler
avec les fichiers pour les changer de place et régénerer la fonte).
Note cependant que le codage cp437 n'inclus pas tous les caractères de
iso-8859-1, notamment manquent quasi toutes les majuscules accentuées, et
les lettres ãõ necessaires en portguais. Tu peux toujours modifier la fonte
pour créer ces caractères manquants.

DB> sur ma slackware 3.2, a aucun moment setfont n'apparait dans les scripts
DB> de demarrage ni dans les ~/profiles et autres ~/.bashrc

Donc c'est la fonte dans la ROM de la carte video qui est utilisée.

DB> et pourtant cette
DB> conversion est effectuee correctement puisque j'ai les accents a l'ecran.

Je suppose que le kernel presupose (à juste titre) que la fonte en ROM video
est au codage cp437.

DB> une fois que je lance "setfont -u none" je perds donc les accents.

Normal, car dès lors que tu charges une fonte t uest supposée savoir ce que
tu veux (au contraire de la fonte en ROM où tu n'as pas le choix).
Autrefois (du temps des 1.2.* et avant) le défaut de setfont était cp437
(et il n'utilsiait pas encore des tables unicode, mais on pouvait
switcher par \e(B et \e(K entre codage cp437 et iso (on peut toujours
d'ailleurs)), mais maintenant le défaut est "straight to font", pas de
conversion.

DB> j'en conclue que linuX utilise la police residente par defaut de
DB> la carte EGA/VGA

DB> [Q] --> comment retrouver cette police ?

il y a un programme (don't j'ai oublié malheureusement le nom) qui est
assez populaire auprès des possesseurs de certaines cartes graphiques chez
qui le switch X11 -> console bousille les fontes; il permet de faire un dump
sur un fichier; il faudrait donc que tu le lances avant de modifier la

```
fonte avec setfont. Le paquetage s'appelle quelque chose comme SVGATextMode
ou quelque chose d'approchant.

 DB>     [Q] --> quelle table appeler pour la remapper et ravoir les accents ?

setfont -u cp437 fontedos

ou sinon, si tu ne veux pas lancer setfont et tu veux donc garder la fonte
de la carte graphique; essaye un "echo -e \033(B" (l'ennuyant c'est que tu
dois le faire dans chaque console)

 DB>                 (en general setfont -u def.uni (par exemple) renvoie une
 DB>                  erreur : PIO_UNIMAPCLR: Invalid argument et je reste
 DB>                  en caracteres "graphiques")

Ah ? Chez moi ça marche; quels kernel et kbd as-tu ?

Sinon je te conseille "cp437" plutôt que "def", ils font _presque_ la même
chose, mais "cp437" est visuellement plus agreable car il remplace les
caractères non présents dans la fonte par les lettres non accentuées si c'est
des lettres, C pour © et R pour ® et par un carré blanc pour le reste, ce qui
permtes de garder le formatage et la lecture agreables; "def" lui ne mets
totu simplement rien pour les caractères manquants, si bien que tu te
trouves avec des trous, ce qui est l'horreur si tu édites un texte par
exemple avec vi, puisque le curseur est affiché plusieurs caractères
à gauche à cause des caractères de largeur nulle :)

Essaye aussi la fonte "t.fnt", elle est assez jolie, totue en rondeurs, je la
trouve réposante, et quand je dois utiliser un PC avec les fontes style
courier par défaut je trouve ça très agressif je me demande comment ils
peuvent travailler avec :)
```

Lire aussi les articles de news : "FAQ - les accents français et Usenet" de F. Yergeau dans groupes fr.usenet.reponses et fr.usenet.8bits "ISO 8859-1 National Character Set FAQ" de mike à vlsivie.tuwien.ac.at dans comp.answers .

6.27 ncurses

Rappellons que ncurses est un librairie C qui permet de manipuler facilement l'affichage en mode console texte : fenêtrage, lignes de saisie, édition de formulaires, couleurs, caractères «ALT» ascii semi-graphique ibm, comme sous dos...

6.27.1 Introduction

Pour les programmes utilisant curses (comme lynx) David Sibley dit :

«La version standard de curses utilise le huitième bit pour la vidéo inversee (voir le flag _STANDOUT défini dans /usr/include/curses.h). Cependant, ncurses semble fonctionner en 8-bits et affiche le iso-latin-8859-1 correctement.»

6.27.2 Les accents

Alors Denis BRAUSSEN (PostOffice à mail.dotcom.fr) répond à cette question :

«Pourquoi en utilisant ncurses 4.2 je n'arrive pas à saisir les accents français dans les formulaires ?»

Remarque : les distrib LinuX sont livrées en général avec la version 1.9.9e de ncurses, or il se trouve que la librairie form de cette version est BUGGÉE (problèmes de rafraichissement des fenêtres).

Ne pas utiliser la 1.9.9e donc si on veut travailler avec 'form' : ce bug est d'ailleurs clairement énoncé sur le site de nucurses, et je m'étonne que ce soit justement cette version qui soit encore livrée sur la plupart des distributions (RedHat 4.2, 5.0, 5.1, SlackWare 3.2, Debian 1.9) en tout cas.

Symptôme : la fonction wgetch() de ncurses renvoie les codes suivants :

- je tape 'é' et wgetch retourne «meta-i» code 233 (decimal)
- " " 'è' " " " «meta-h» code 234 (decimal)
- etc

C'est normal avec les polices iso-8859-1 !

Il faut installer les "locale" (ben oui, allez savoir pourquoi, sur la slackware 3.2 elles ne sont pas installées !).

Note : les 'locales' sont des fichiers qui vont modifier le comportement de certains programmes pour qu'ils s'adaptent aux «spécificités culturelles du pays ».

- récupererer :
- pour la libc5 la collection de locales POSIX : ftp ://ftp.lip6.fr/pub/linux/GCC/WG15-collection.linux.tar.gz
- pour la libc6, elle est dans l'ajout «localedata» : ftp ://ftp.lip6.fr/pub/gnu/glibc-localedata-2.0.6.tar.gz
- copier dans le répertoire /usr/share/locale les fichiers fr_FR et en_DK (qui est inclus dans fr_FR)
- créer le répertoire /usr/share/i18n/charmap et copier le fichier ISO_8859-1 :1987
- faire man localedef et lancer la commande :

  ```
  localedef -f ISO_8859-1:1987 -i fr_FR fr
  ```

  Ceci créera l'entrée fr dans /usr/share/locale soit :

  ```
  /usr/share/locale/fr
  ```

  Ce répertoire devrait maintenant contenir les fichiers

  ```
  LC_COLLATE
  LC_CTYPE
  LC_MESSAGES
  LC_MONETARY
  LC_NUMERIC
  LC_TIME
  ```

- lire le fichier «locale.fr» que l'on peut trouver sur le serveur ftp.lip6.fr (chercher sous doc, linux...) et dont la traduction française a été faite par Éric DUMAS (hélas ce doc ne traite principalement que de LC_MESSAGES. À quand la suite Éric ? ;-))
- compléter éventuellement l'intitialisation des variables du shell comme décrit dans la section sur bash ou tcsh
- dans un programme, utilisez la fonction setlocale() (man setlocale.3) pour fixer les paramêtres locaux CAR le noyau linux initialise toujours cette fonction avec l'argument une locale «C» (POSIX) donc sans accents !

  ```
  setlocale(LC_CTYPE,"fr_FR");
  ```

- Hop ! ! On a nos accents dans ncurses, le plus simplement du monde.

6.28 Divers

Pour les programmes utilisant groff (comme man), utilisez le avec :

```
-Tlatin1
```

au lieu de :

```
-Tascii
```

Les vieilles versions de man utilisent aussi col, et le point suivant s'applique aussi.

Une belle discussion sur le thème de l'ISO-8859-1 et sur «comment manipuler les caractères 8-bits» est disponible sur `ftp ://grasp.insa-lyon.fr/pub/faq/fr/accents`

Pour perl, si tout d'un coup après une mise à jour il raconte :

```
(guylhem@barberouge:guylhem)$ perl
perl: warning: Setting locale failed for the categories:
        LC_CTYPE LC_COLLATE
perl: warning: Please check that your locale settings:
        LC_ALL = "fr_FR",
        LC_CTYPE = "ISO-8859-1",
        LC_COLLATE = (unset),
        LANG = "fr"
    are supported and installed on your system.
perl: warning: Falling back to the "C" locale.
(guylhem@barberouge:guylhem)$
```

Essayez cette solution de Thomas Parmelan (Thomas.Parmelan à efrei.fr) :

En libc6 le format des fichiers binaires de définition des locales a (encore) changé : il faut les régénérer à partir des sources (WG15collection.tar.tz sur ftp.lip6.fr :/pub/linux/GCC par exemple) et des programmes "locale" et "localedef" qui sont fournis dans les sources de la libc.

Sinon tu peux également essayer ceci (c'est comme cela que ça fonctionne chez moi en libc-5.4.x) : »

```
unset LANG
unset LC_CTYPE
unset LC_COLLATE
export LC_ALL=fr_FR.ISO-8859-1
```

7 Réglage du clavier pour les applications X

7.1 Les xterminaux (xterm, nxterm, rxvt...)

Les faire fonctionner avec toutes les touches standard du clavier 102 touches relève du miracle, quant aux accents, seule une recompilation vous permettra de les utiliser !

Pour xterm, essayez de mettre dans votre `.Xdefaults` :

```
XTerm*eightBitInput: true
XTerm*eightBitOutput: true
*customization: -color
XTerm*VT100*Translations: #override\n\
XTerm*pointerColor: red
None<Key>Begin: string(0x1b) string("[1~")\n\
```

```
None<Key>End: string(0x1b) string("[4~")\n\
None<Key>Prior: scroll-back(1,halfpage)\n\
None<Key>Next: scroll-forw(1,halfpage)\n\
Shift<Key>Prior: scroll-back(1,page)\n\
Shift<Key>Next: scroll-forw(1,page)
```

Pablo Saratxaga conseille de rajouter dans `.inputrc` :

```
# pour rxvt
"\e[7":beginning-of-line
"\e[8":end-of-line
# pour xterm
"\e[H": beginning-of-line
"\e[F": end-of-line
```

Benjamin Dauvergne pour sa part ajoute cela au /.Xdefaults pour bénéficer d'un xterm au menu des fontes francisé :

```
XTerm*fontMenu*fontdefault*Label:      Default
XTerm*VT100*font:              9x15
XTerm*fontMenu*font1*Label:    Illisible
XTerm*VT100*font1:             nil2
XTerm*fontMenu*font2*Label:    Minuscule
XTerm*VT100*font2:             5x7
XTerm*fontMenu*font3*Label:    Petit
XTerm*VT100*font3:             6x10
XTerm*fontMenu*font4*Label:    Normal
XTerm*VT100*font4:             fixed
XTerm*fontMenu*font5*Label:    Moyen
XTerm*VT100*font5:             7x13
XTerm*fontMenu*font6*Label:    Tres grand
XTerm*VT100*font6:             10x20
```

Mais ces deux solutions ne vous garantissent pas de résoudre tous les problèmes de rxvt et de xterm ...

Pour les résoudre sans créer de fichiers dans `/usr/X11R6/lib/X11/app-defaults`, je tiens sinon à votre disposition un xiterm, un terminal X supportant *TOUTES* les touches des claviers standard, avec Home, End, Page_up (...) et même le pavé numérique, fonctionnant comme en mode console, i.e. Shift-Page_up permettant de faire défiler une page, voire mieux : le pavé numérique reconnaissant enfin la différence entre Verr. Num allumé et Verr. Num éteint.

X international terminal (xiterm pour les intimes) est bien sur sous GPL et mis à jour pour suivre parallèlement l'évolution de rxvt.

Sur par exemple http ://www.linux-france.com/prj/xiterm ou sur ftp ://sunsite.unc.edu/pub vous trouverez donc xiterm, qui remplace votre xterm classique en y apportant la couleurs, les accents et les touches étendues.

Pour l'installer, compilez-le, devenez root et tapez :

```
rm /usr/X11R6/bin/xterm
rm /usr/X11R6/lib/X11/app-defaults/xterm
rm /usr/X11R6/lib/X11/app-defaults/rxvt
rm /usr/X11R6/lib/X11/app-defaults/Xterm
rm /usr/X11R6/lib/X11/app-defaults/XTerm
rm /usr/X11R6/lib/X11/app-defaults/Rxvt
rm /usr/X11R6/lib/X11/app-defaults/RXvt
cp xiterm /usr/local/bin
```

```
ln -sf /usr/local/bin/xiterm /usr/X11R6/bin/xterm
ln -sf /usr/local/bin/xiterm /usr/X11R6/bin/rxvt
```

7.2 Les applications Motif

Là, je n'ai malheureusement aucun remède miracle !

Utilisez les possibilités du fichier french au lieu des touches mortes (AltGr (voyelle) : voyelle accent circonflexe...) !

7.3 Le manuel

Pablo Saratxaga tient à votre disposition un xman modifié pour qu'il supporte la variable «LANG» et puisse afficher les pages en d'autres langues qu'anglais.

Il est disponible sur :

`ftp.lip6.fr :/pub/linux/distributions/redhat-contrib/SRPMS/xman-3.1.6-3.src.rpm`
`ftp.lip6.fr :/pub/linux/distributions/redhat-contrib/i386/xman-3.1.6-3.i386.rpm »`

7.4 NumLock

Frédéric Petit nous dit que «j'ai découvert que si l'on met ServerNumLock dans la section keyboard de XF86Config, ça marche très bien ; pour mes propres besoins, j'ai commenté cette ligne et j'ai mis sur la ligne en-dessous Xleds 2, comme ça, sous X11, je suis automatiquement en mode NumLock pour le pad numérique et la led correspondante est allumée ...»

7.5 XDM

Frédéric Petit nous conseille d'utiliser ce /etc/X11/xdm/Xresources afin de ne rencontrer aucun problème avec nos touches «spéciales» (Home, End ...) sous X.

```
xlogin*login.translations: #override\
      Ctrl<Key>R: abort-display() \n\
      <Key>BackSpace: delete-previous-character() \n\
      <Key>Home: move-to-begining() \n\
      <Key>Delete: delete-character() \n\
      <Key>End: move-to-end() \n\
      <Key>Left: move-backward-character() \n\
      <Key>Return: set-session-argument() finish-field() \n\
      <Key>Right: move-forward-character()
```

8 Remerciements

Ce document ne serait pas ce qu'il est sans l'aide de Nat Makarévitch (nat à nataa.fr.eu.org) qui l'a relu et m'a aidé à le modifier de nombreuses fois.

Ce document est un hybride nait des French HOWTO 1 et Francophones-HOWTO qui a rassemblé toutes leurs informations en un seul HOWTO destiné à la communauté linux francophone.

La section «Configurer les applications en mode texte» s'inspire pour sa part des paragraphes sur emacs, zsh, tcsh, flex, less, rlogin, joe, col et TeX du Guide du Rootard (le second HOWTO destiné à la communauté linux francophone

Rappelons que les sites officiels du Guide du Rootard sont :

`http ://www.freenix.fr/linux/Guide`

et

`ftp.lip6.fr :/pub/linux/french/docs/`

Remercions enfin par ordre alphabétique ces contribueurs :

Jean-Michel Antoine
Michel Billaud
Stephane Bortzmeyer
Rémy Card
Xavier Cazin
Laurent Chemla
René Cougnenc
Bruno Cornec
Yann Dirson
Éric Dubreuil
Éric Dumas
Pierre Ficheux
Laurent Frigault
Frederic Lepied
Stephane Marzloff
Marc Merlin
Thomas Parmelan
Thomas Quinot
David Robert
Olivier Robert
Benjamin Ryzman
Pablo Saratxaga
Julien Simon
J.M. Vansteene

Surtout, si je vous oublie, signalez-le moi ! De même si vous ne voulez plus figurer dans cette liste...

Envoyez moi vos suggestions et modifications (guylhem à oeil.qc.ca), je me ferai un plaisir de les intégrer dans les prochaines versions !

Signalez-moi aussi les erreurs : cet HOWTO est en perpétuel changement, avec des nouveautés incluses à chaque version ; des erreurs peuvent facilement s'y glisser !

9 Fichiers inclus dans ce HOWTO

Comme signalé plus haut, cette section ne contient que french (pour X window) et french.map (pour linux-console).

Voici french (pour X11, à placer dans /usr/X11R6/lib/X11/xkb/symbols) :

```
// French keyboard description, (c) Guylhem Aznar, July 1997
// Released under GNU GENERAL PUBLIC LICENSE Version 2
```

```
// If you haven't received it, this license is available at :
// Free Software Foundation, Inc.  675 Mass Ave, Cambridge, MA 02139, USA

partial hidden alphanumeric_keys modifier_keys
xkb_symbols "basic" {

// Description *TOTALE* d'un clavier français, sans le comparer à un
// clavier américain par défaut : on y gagne plus de flexibilité, ce
// modèle devient plus personalisable ...

// Les touches é, è, ç, à & ù, présentes en mode minuscule (i.e. sans shift)
// ne sont pas disponibles en mode majuscule car elles sont respectivement
// remplacées par 2, 7, 9, 0 & % .
// Pour obtenir les majuscules accentuées É, È, Ç, À & Ù, vous pouvez soit
// utiliser AltGr + Shift (minuscule accentuée), soit utiliser les accent
// dits "morts", i.e. fonctionnant comme l'accent circonflexe & le tréma .
// Sont disponibles :  AltGr (1) : accent aigü, AltGr (7) : accent grave,
// AltGr (^) : cédille ; pour s'en servir, procéder comme avec l'accent
// circonflexe & le tréma : ex : AltGr (1) + e : é ; AltGr (1) + e : É ...
// La tilde (~) n'étant pas employée sur des lettres en français, elle reste
// en AltGr (2) une touche normale .
// Une tilde "morte" (avec la cédille "morte") est toutefois disponibles sur la
// touche accent circonflexe & tréma en AltGr + Shift .

// Au cas où les dead_keys ne fonctionnent pas (certains programmes les
// refusent), vous pouvez toujours utiliser pour obtenir les voyelles à accent
// circonflexe & à tréma :
// AltGr (voyelle) : voyelle-minuscule à accent circonflexe
// AltGr + Shift (voyelle) : voyelle-minuscule à tréma
// AltGr (en dessous de la voyelle) : voyelle-majuscule à accent circonflexe
// AltGr + Shift (en dessous de la voyelle) : voyelle majuscule à tréma

// Plusieurs lettres étrangères, utilisées comme signes mathématiques ou
// physiques, abbréviations (...) ont été rajoutées en AltGr :
// T->Phi;, r->(TM), y->yen, c-> et ©, f->señora, m->señor,
// g->‘, h->’, ù->±, x-> s-> , z-> , w->Delta;, b->

// La correspondance essaye de se faire la plus naturelle possible, en mettant
// ces touches là où l'on s'attend le plus de les trouver ...
// mais elle est loin d'être parfaite :^)

// Modifiez donc à loisir tous ces AltGr, et si vous pensez avoir trouvé la
// configuration ultime permettant d'accéder en mieux à toutes ces touches,
// mailez moi !
// Vôtre arrangement, s'il est vraiement plus pratique, sera inclu dans la
// prochaine version de la carte de correspondance du clavier !

    name[Group1]= "French";
    key <ESC>   {       [          Escape                        ]       };

// Deux possibilitées pour cette touche "au carré" (²) :
// * un mode mathématique où elle renvoie "au carré" en mode normal, "au cube"
// si elle est Shiftée et "puissance 1" en Alt-gr (pour parler par exemple de
// m.s-¹ ...)
// Toutefois, accent circonflexe suivi de 1, 2, 3, -, renvoyant respectivement
```

```
// ¹, ², ³ et —, cette touche est inutile !

// * un mode littéraire est donc proposé par défault, pour disposer des
// guillemets français « & » ; le ² et le ³ restant disponibles
// respectivement en AltGr et AltGr + Shift .

//   key <TLDE> {       [     twosuperior,   threesuperior      ],
//                      [     onesuperior                       ]       };

    key <TLDE>  {       [   guillemotleft,  guillemotright      ],
                        [     twosuperior,   threesuperior      ]       };

    key <AE01>  {       [       ampersand,              1       ],
                        [      dead_acute                       ]       };
    key <AE02>  {       [          eacute,              2       ],
                        [      asciitilde,          Eacute      ]       };
    key <AE03>  {       [        quotedbl,              3       ],
                        [      numbersign                       ]       };
    key <AE04>  {       [      apostrophe,              4       ],
                        [       braceleft                       ]       };
    key <AE05>  {       [       parenleft,              5       ],
                        [     bracketleft                       ]       };
    key <AE06>  {       [           minus,              6       ],
                        [             bar                       ]       };
    key <AE07>  {       [          egrave,              7       ],
                        [      dead_grave,          Egrave      ]       };
    key <AE08>  {       [      underscore,              8       ],
                        [       backslash                       ]       };
    key <AE09>  {       [        ccedilla,              9       ],
                        [     asciicircum,        Ccedilla      ]       };
    key <AE10>  {       [          agrave,              0       ],
                        [              at,          Agrave      ]       };
    key <AE11>  {       [      parenright,          degree      ],
                        [    bracketright                       ]       };
    key <AE12>  {       [           equal,            plus      ],
                        [      braceright                       ]       };
    key <BKSP>  {       [       BackSpace                       ]       };
    key  <TAB>  {       [             Tab,    ISO_Left_Tab      ]       };
    key <AD01>  {       [               a,               A      ],
                        [     acircumflex,      adiaeresis      ]       };
    key <AD02>  {       [               z,               Z      ],
                        [           aring,           Aring      ]       };
    key <AD03>  {       [               e,               E      ],
                        [     ecircumflex,      ediaeresis      ]       };
    key <AD04>  {       [               r,               R      ],
                        [      registered                       ]       };
    key <AD05>  {       [               t,               T      ],
                        [           thorn,           THORN      ]       };
    key <AD06>  {       [               y,               Y      ],
                        [             yen                       ]       };
    key <AD07>  {       [               u,               U      ],
                        [     ucircumflex,      udiaeresis      ]       };
    key <AD08>  {       [               i,               I      ],
                        [     icircumflex,      idiaeresis      ]       };
    key <AD09>  {       [               o,               O      ],
```

```
                    [     ocircumflex,      odiaeresis      ]       };
key <AD10>  {       [               p,               P      ],
                    [       paragraph                       ]       };
key <AD11>  {       [ dead_circumflex,  dead_diaeresis      ],
                    [    dead_cedilla,      dead_tilde      ]       };
key <AD12>  {       [          dollar,        sterling      ],
                    [        currency                       ]       };
key <RTRN>  {       [          Return                       ]       };
key <AC01>  {       [               q,               Q      ],
                    [     Acircumflex,      Adiaeresis      ]       };
key <AC02>  {       [               s,               S      ],
                    [          oslash,        Ooblique      ]       };
key <AC03>  {       [               d,               D      ],
                    [     Ecircumflex,      Ediaeresis      ]       };
key <AC04>  {       [               f,               F      ],
                    [     ordfeminine                       ]       };
key <AC05>  {       [               g,               G      ],
                    [      quoteleft                        ]       };
key <AC06>  {       [               h,               H      ],
                    [     quoteright                        ]       };
key <AC07>  {       [               j,               J      ],
                    [     Ucircumflex,      Udiaeresis      ]       };
key <AC08>  {       [               k,               K      ],
                    [     Icircumflex,      Idiaeresis      ]       };
key <AC09>  {       [               l,               L      ],
                    [     Ocircumflex,      Odiaeresis      ]       };
key <AC10>  {       [               m,               M      ],
                    [      masculine                        ]       };
key <AC11>  {       [          ugrave,          percent      ],
                    [       plusminus,           Ugrave      ]       };
key <BKSL>  {       [        asterisk,               mu      ],
                    [     Greek_OMEGA                       ]       };
key <LFSH>  {       [          Shift_L                      ]       };
key <LSGT>  {       [           less,           greater      ],
                    [             bar                       ]       };
key <AB01>  {       [               w,               W      ],
                    [             eth,             ETH      ]       };
key <AB02>  {       [               x,               X      ],
                    [              ae,              AE      ]       };
key <AB03>  {       [               c,               C      ],
                    [            cent                       ]       };
key <AB04>  {       [               v,               V      ],
                    [       copyright                       ]       };
key <AB05>  {       [               b,               B      ],
                    [          ssharp                       ]       };
key <AB06>  {       [               n,               N      ],
                    [         notsign                       ]       };
key <AB07>  {       [           comma,         question      ],
                    [    questiondown                       ]       };
key <AB08>  {       [       semicolon,           period      ],
                    [        multiply                       ]       };
key <AB09>  {       [           colon,            slash      ],
                    [        division                       ]       };
key <AB10>  {       [          exclam,          section      ],
                    [      exclamdown                       ]       };
```

```
    key <RTSH>  {       [         Shift_R                     ]       };

    key <LCTL>  {       [       Control_L                     ]       };
//    key <LALT>        {       [           Alt_L,          Meta_L      ]       };
    key <SPCE>  {       [           space                     ]       };
//    key <RALT>        {       [           Alt_R,          Meta_R      ]       };
    key <RCTL>  {       [       Control_R                     ]       };

    key <FK01> {        [        F1             ]       };
    key <FK02> {        [        F2             ]       };
    key <FK03> {        [        F3             ]       };
    key <FK04> {        [        F4             ]       };
    key <FK05> {        [        F5             ]       };
    key <FK06> {        [        F6             ]       };
    key <FK07> {        [        F7             ]       };
    key <FK08> {        [        F8             ]       };
    key <FK09> {        [        F9             ]       };
    key <FK10> {        [        F10            ]       };
    key <FK11> {        [        F11            ]       };
    key <FK12> {        [        F12            ]       };

    key <PRSC> { type= "PC_SYSRQ", symbols[Group1]= [ Print, Execute ] };
    key <SCLK> {  [  Scroll_Lock        ]       };
    key <PAUS> { type= "PC_BREAK", symbols[Group1]= [ Pause, Break ] };
    key  <INS> {  [  Insert             ]       };
    key <HOME> {        [  Home                 ]       };
    key <PGUP> {        [  Prior                ]       };
    key <DELE> { [ Delete               ]                       };
    key  <END> {        [  End                  ]       };
    key <PGDN> {        [  Next                 ]       };

    key   <UP> {        [  Up                   ]       };
    key <LEFT> {        [  Left                 ]       };
    key <DOWN> {        [  Down                 ]       };
    key <RGHT> {        [  Right                ]       };

    key <NMLK> {        [  Num_Lock, Pointer_EnableKeys         ]       };
    key <KPDV> {        [  KP_Divide            ]       };
    key <KPMU> {        [  KP_Multiply          ]       };
    key <KPSU> {        [  KP_Subtract          ]       };

    key  <KP7> {        [  KP_Home,     KP_7    ]       };
    key  <KP8> {        [  KP_Up,       KP_8    ]       };
    key  <KP9> {        [  KP_Prior,    KP_9    ]       };
    key <KPAD> {        [       KP_Add          ]       };

    key  <KP4> {        [  KP_Left,     KP_4    ]       };
    key  <KP5> {        [  KP_Begin,    KP_5    ]       };
    key  <KP6> {        [  KP_Right,    KP_6    ]       };

    key  <KP1> {        [  KP_End,      KP_1    ]       };
    key  <KP2> {        [  KP_Down,     KP_2    ]       };
    key  <KP3> {        [  KP_Next,     KP_3    ]       };
    key <KPEN> {        [       KP_Enter        ]       };
```