# Le Mini-Howto Mac Terminal

Robert Kiesling, traduit par Olivier Tharan, `olive@laria.u-picardie.fr` v1.4, 9 novembre 1997

Ce mini Howto décrit la 1002ème utilisation d'un Macintosh mort (hé hé) : comment configurer le Mac pour en faire un terminal pour Linux. Des configurations utilisant getty et le programme de terminaux kermit sont décrites, ainsi que l'utilisation de kermit en réseau point à point entre Linux et un Macintosh. Ce document peut être reproduit librement, en totalité ou en partie, à la condition que toute utilisation soit conforme à la note générale de copyright du Projet de Documentation Linux. Voyez le fichier COPYRIGHT pour plus de détails. Envoyez toutes les plaintes, suggestions, erreurs et autres à *kiesling@terracom.net* `mailto:kiesling@terracom.net` , pour que je puisse garder ce document aussi complet et à jour que possible.

## Contents



# 1 Introduction

Ce mini Howto devrait vous donner quelques idées Super Top pour faire fonctionner votre Macintosh avec Linux. Malheureusement, j'ai été très occupé et je n'ai donc pas pu inclure la moitié de ce que je prévoyais d'inclure, comme l'utilisation de MacTCP et d'OpenTransport pour vous connecter à votre machine Linux par l'intermédiaire d'une ligne PPP. Ceci devra attendre une version future.

Ce mini Howto ne couvre pas non plus la mise en place sur un réseau LocalTalk ou AppleTalk. J'explorerai peut-être ces thèmes s'il y a suffisamment d'intérêt dans, disons, l'impression sur une imprimante LaserWriter à partir de Linux. Sinon, il me semble que de telles applications, causant plus de tracas qu'il n'en faut (sans parler du prix), ne tiennent pas dans le cadre de ce document.

Je ne compte pas parler de MkLinux non plus dans ce document. C'est bien mieux documenté autre part.

Si vous avez donc des idées à propos de ce document, laissez-moi un mot à l'adresse électronique ci-dessus. Les deux systèmes renferment une bonne partie de l'état d'esprit du débutant, ainsi qu'une prouesse technique, et à mon sens ils ne communiquent pas assez entre eux.

# 2 Établissement d'une liaison série

Pour établir une liaison série entre un Mac et une machine Linux, vous avez besoin, du côté Linux, soit d'un câble série DB9 femelle vers DB25 mâle ou un câble série DB25 femelle vers DB25 mâle, selon votre port série. Du côté du Macintosh, vous avez besoin d'un câble modem à grande vitesse DIN9 vers DB25 mâle.

Assurez-vous que le câble est bien un câble à "grande vitesse", car certains câbles Macintosh anciens sont configurés avec des lignes de prise de contact (handshaking) liées, ce qui les rend inutiles pour des connexions série à grande vitesse.

Vous aurez aussi besoin d'un adaptateur null modem, disponible chez tout revendeur informatique, et d'un changeur de genre série DB25 femelle vers DB25 femelle pour relier les deux câbles série.

J'ai entendu dire que les câbles d'imprimante pour Mac sont en fait des câbles null modem déguisés, mais je ne peux pas le confirmer. Certains d'entre eux sont de toute façon des DIN9 vers DIN9, et en connecter un sur une liaison série engendrerait plus de tracas que nécessaire.

Si cela ressemble à de l'Hébreu pour vous, lisez le Howto Serial pour plus de détails sur la configuration des câbles RS 232 et sur les protocoles de transmission de données.

Avant de connecter le Mac et la machine Linux, vous devriez déterminer si vous avez un port série en état de marche sur les deux machines, soit en connectant un modem et en appelant un autre ordinateur avec `minicom` (Linux), ZTerm (Mac), `kermit` (l'un ou l'autre), ou tout programme de communication de votre choix.

La dernière version de minicom est disponible à *sunsite.unc.edu/pub/Linux/apps/serialcomm/dialout* `ftp://sunsite.unc.edu/pub/Linux/apps/serialcomm/dialout` et sur les sites miroir.

ZTerm est un programme de communication complet et facile à utiliser. Malheureusement, c'est un shareware. Une version à jour est disponible à *mac.archive.umich.edu* `ftp://mac.archive.umich.edu` et de telles archives.

Le programme `kermit` a été porté sur tous les ordinateurs et systèmes d'exploitation qui existent. Les archives sont situées à *ftp.columbia.edu/kermit* `ftp://ftp.columbia.edu/kermit` .

Vous devriez considérer sérieusement l'utilisation de `kermit` sur les deux machines au moins à ce point, parce que 1) il est libre (bien qu'il ne soit pas couvert par la Licence Publique Générale de la Free Software Foundation) ; et 2) il est bien moins gênant d'avoir `kermit` sur les deux machines que d'avoir deux programmes de communication complètement différents.

Si vous avez un autre moyen de déterminer si les ports série des deux machines sont fonctionnels, n'hésitez pas à l'utiliser. Ce qui est important est de s'assurer que les deux machines possèdent un port série en état de marche.

La connexion série réelle devrait être facile, si l'on a suivi les indications ci-dessus. Si ce n'est pas le cas, la connexion ressemble à ceci :

```
PC Linux    Câble série    Adap-   Changeur  Câble série     Mac
---------    DB9 ou DB25    tateur  de genre  DIN9 vers     --------
|       |   vers DB25 mâle |     |  |     |  DB25 Mac      |      |
|       |------------------|     |--|     |-----------------|      |
|       |                  |     |  |     |                 |      |
---------                  Null                             --------
                           Modem
```

# 3 Connexion client-serveur avec kermit

C'est la configuration la plus fluctuante parmi celles décrites ici. Elle demande le moins de configuration système, bien qu'à l'usage ce soit le système le plus difficile à utiliser parmi ceux décrits ici.

En bref, vous démarrez `kermit` à la fois sur la machine Linux et sur le Mac et vous placez l'un d'entre eux en mode serveur. Que l'une soit le client et l'autre le serveur n'a aucune importance, parce que c'est une connexion point à point. Cependant, le `kermit` de Linux peut tirer partie des possibilités de script avancées de Linux, et il semble donc logique (pour moi au moins) de donner au `kermit` côté Linux la fonction de serveur, car cette tâche sera plus rapidement automatisée.

Vous devez vous assurer que `kermit` est installé correctement à la fois sur le Mac et sur le PC Linux. Suivez les instructions des distributions kermit respectives. Sur la machine Linux tapez `kermit` à l'invite du shell pour le démarrer. Vous aurez peut-être besoin des droits root pour positionner la vitesse du port et de la connexion.

`Kermit`, dans les versions POSIX récentes pour Unix, supporte des vitesses de connexion jusqu'à 115 Kbps. Les versions les plus récentes pour Macintosh supportent des vitesses de port série allant jusqu'à 57,6 Kbps. Ceci devrait être plus que suffisant pour toute application basique de type terminal, mais si vous avez besoin d'une connexion à plus grande vitesse, c'est raté en ce qui concerne `kermit` et les liaisons série. Cependant, `kermit` permet des communications aisées sur une liaison TCP/IP, mais je n'ai pas pu tester cette possibilité. Regardez les autres solutions dans les sections suivantes. Rappelez-vous simplement, surtout du côté du Mac, d'utiliser avec `kermit` un port différent pour vos connexions série que pour vos connexions TCP/IP, car `kermit` sur le Mac mobilisera de manière brutale un port série en cours d'utilisation.

Avec ceci en tête, votre fichier `.kermrc` devrait contenir quelque chose comme ceci :

```
echo Lecture du fichier global /usr/local/bin/ckermit.local.ini....
set prompt Chanel3 >
set line /dev/ttyS0
set baud 38400
set send packet-length 2000
set receive packet-length 2000
set block 3
set file type binary
```

Puis, dans votre fichier `~/.kermrc`, vous auriez une ligne telle que

```
take /usr/local/bin/ckermit.local.ini
```

Du côté du Macintosh, positionnez les mêmes paramètres de communication pour les bits par seconde, les bits de stop, la parité et la longueur d'un mot. Certaines versions anciennes de Kermit Mac ne supportent pas les paquets de 2 k, par conséquent vous aurez peut-être besoin d'indiquer une taille de paquet plus petite. Cependant, `kermit` positionne la longueur du paquet de communication selon le paramètrage de longueur de paquet reçu, et donc vous aurez besoin de mettre une taille de paquet plus petite du côté Linux également.

Pour communiquer effectivement sur la liaison, vous devez entrer en mode serveur soit sur le Mac soit sur Linux. Lequel n'a pas d'importance. Voyez la documentation `kermit` pour avoir des détails sur le mode serveur.

## 3.1 Ressources Macintosh

Voici l'une des rares applications de `kermit` où le positionnement d'un type de fichier `texte` pour les transferts est utile. C'est parce que les fichiers Macintosh ont deux parties : la **partie données** et la **partie**

**ressources**. La partie données correspond à ce que nous appellerions un fichier dans le monde Linux : ce sont les données réelles. La partie ressource contient des images pour les icônes, des jeux de claviers, des spécifications de polices, et ainsi de suite. Si vous transférez un fichier de Linux vers le Mac, le fichier ne sera pas reconnu comme un fichier texte par le Mac, si vous utilisez le mode binaire.

En transférant des fichiers binaires entre les deux systèmes, vous devriez utiliser le format BinHex Macintosh `.hqx`, qui est un codage sur 7 bits d'un fichier de données en 8 bits. Des utilitaires Mac comme BinHexer ou StuffIt convertiront le fichier en sa forme binaire.

Si vous avez un fichier texte qui devient par inadvertance un fichier de données seules sur le Mac, il est probable qu'il n'apparaîtra jamais dans une liste de boîte de dialogue d'ouverture de fichier. Ce que vous devrez faire est ouvrir le fichier avec ResEdit, disponible à `mac.archive.umich.edu`. ResEdit vous dira que le fichier que vous ouvrez n'a pas de partie ressources et vous demandera alors si vous voulez en ajouter une. Vous devriez répondre "Yes" (oui) à cette question. Vous pourrez alors éditer les champs Type et Creator du fichier en sélectionnant l'option "Open Special" du menu Fichier. Tous les fichiers texte Macintosh sont du type `TEXT`, remplacez donc les points d'interrogation dans la boîte de Texte avec cela. Le code Creator dépend de votre éditeur de texte ou de votre traitement de texte. Chacun est unique, justement, et c'est ainsi que le Mac identifie des applications différentes. Le code Creator pour GNU Emacs sur le Mac est `EMAC`, par exemple. Si vous avez un doute sur le code du créateur de votre éditeur de texte ou traitement de texte, utilisez `ttxt`, qui est le code créateur de TeachText (qui est l'équivalent Mac de `EDLIN.EXE`.) Votre traitement de texte ou éditeur de texte réel pourra alors traduire le fichier de TeachText en son type natif.

On peut faire bien d'autres choses avec TeachText, ça vaut donc la peine de le garder en permanence sur votre Mac. Le livre *Voodoo Mac*, par Kay Yarborough Nelson, est une bonne source de trucs pour Macintosh qui ont fait leur preuve et qui utilisent ResEdit, TeachText, le Finder et d'autres programmes un peu oubliés.

# 4 Se logger grâce à kermit

Configurer Linux pour utiliser le Mac comme un terminal de `login:` est encore plus facile. `kermit` est idéal pour cela, parce que c'est l'un des rares programmes de communication libres qui fournit une émulation VT100/120/220 crédible.

De manière générale, ce que vous voudrez faire est lancer `kermit` sur le Macintosh comme dans la section précédente, mais plutôt que de taper des commandes sur le serveur, vous entrez en mode `connect`. C'est le mode normal d'émulation de terminal qu'utilisent de toute façon la plupart des gens.

Du côté Linux, la ligne série doit être configurée grâce à un `getty` pour démarrer un shell `login:`. À cette fin, vous devez dire à `init` qu'un terminal est attaché à la ligne série. Dans votre fichier `/etc/inittab` vous avez besoin d'une ligne comme celle-ci :

```
T1:23:respawn:/sbin/getty -L ttyS0 9600 vt100
```

Assurez-vous de mettre le périphérique série approprié à la place de `/dev/ttyS0` et la vitesse correcte pour le port à la place de `9600` dans la ligne de commande ci-dessus.

Cette commande indique à `getty` de démarrer `login` (le paramètre `-L`) sur l'affichage du terminal, et quand la période de login est terminée (timeout), de redémarrer (respawn, relancer) le programme de login jusqu'à ce que quelqu'un se logge. Si aucun périphérique n'est connecté à la ligne série, ou si la connexion est mauvaise, vous pourrez voir un message sur la console du système tel que : `/dev/ttyS0 respawning too fast: disabling for 5 minutes` (/dev/ttyS0 se relance trop vite, je l'arrête pendant 5 minutes). Si c'est le cas, vous pouvez revenir à un comportement normal en (en tant que root) tuant avec `kill` le processus `getty`, ou en utilisant la commande `init q`. Les deux ont l'effet de relancer le(s) processus `getty`. Si tout est en ordre, vous devriez voir la bannière Linux et l'invite de login sur la fenêtre `kermit` du Mac. C'est tout.

De plus, si vous utilisez autre chose que le `getty` de base, comme `getty_ps`, la commande ci-dessus sera légèrement différente. Ce qu'il convient de se rappeler est que tout ce qui est à droite de `/sbin/getty` est un argument de `getty` et non d'`init`. Vous devriez jeter un coup d'oeil aux pages de manuel de `getty`, `init` et `inittab` si vous vous posez des questions sur la mise en oeuvre de `init` et de `getty`.

Le Howto Serial fournit des détails utiles sur la manière de configurer `/etc/inittab` pour `getty_ps`, si c'est ce que votre système utilise.

Pour transférer des fichiers entre le Macintosh et la machine Linux, vous pouvez (à travers le Kermit du Mac) lancer la commande `kermit -x` pour démarrer le `kermit` de Linux en mode serveur. Vous pouvez alors utiliser les commandes normales de transfert de fichiers pour envoyer des fichiers sur la ligne série. Il est utile de positionner une invite dans votre `~/.kermrc` avec une ligne telle que

```
set prompt Linux-kermit >
```

Sinon, se rappeler sur quelle machine on est peut vite devenir déroutant.

### 4.1 Autres programmes d'émulation de terminal sur Mac

Cette méthode devrait fonctionner de manière équivalente sur n'importe quel autre programme d'émulation de terminal pour Mac. Si vous avez ZTerm, vous pouvez utiliser `rz` et `sz` sur la machine Linux pour transférer des fichiers grâce au protocole ZModem. Si Microphone Lite était livré avec votre modem-fax, cela fonctionne à l'identique, bien que sans les possibilités évoluées de script et de configuration de `kermit`.

## 5 Conclusion

Si vous avez des questions à propos de tout ceci, ou des suggestions concernant l'évolution future de la connectivité par liaison série entre un Mac et Linux, n'hésitez pas à m'envoyer un mot à `kiesling@terracom.net`.