

TOMSRTBT, utilisation et personnalisation

Jean-Marc LICHTLE

22 novembre 2003

Table des matières

1	Introduction	1
2	Personnalisation	2
2.1	Sélection du clavier	2
2.2	Fichiers de configuration	2
2.3	Retour à la sélection du clavier	3
2.4	Paramétrage de la connexion réseau	4
3	Prise de contrôle à distance, telnet	5
3.1	Démon telnetd	5
3.2	Client telnet	6
4	Transfert ftp	6
4.1	Client ftp	6
4.2	Serveur ftp	7
4.2.1	Transfert de fichiers à l'unité	7
4.2.2	Copie d'un disque complet	7
5	Ajout de fonctionnalités supplémentaires	8
5.1	Connecter un lecteur ZIP	8

1 Introduction

Tomsrtbt est une microdistribution LINUX tenant sur une seule disquette. Évidemment avec une telle compacité il ne faut pas s'attendre à trouver X Windows avec la dernière mouture de KDE. Mais tout de même, on arrive à trouver les outils de base qui permettraient, le cas échéant, de se dépanner en présence d'une machine récalcitrante, de tester un réseau ou toute autre activité connexe.

La dernière mouture de la distribution est facile à trouver sur le site de Tom OEHSER :

<http://www.toms.net/rb/>

Son installation est décrite dans un article de Marc sur Lea-Linux.org, à l'époque rédigé pour la version 1.7.361 mais qui reste d'actualité pour la 2.0.103, la dernière mouture en date à l'heure ou j'écris ces lignes.

J'ai cherché longtemps une distribution qui soit plus performante, je dois avouer que Tomsrtbt est simplement un modèle du genre, à tel point que d'autres produits, peut-être plus faciles d'accès, avouent sans fausse honte qu'ils en sont largement inspirés.

2 Personnalisation

2.1 Sélection du clavier

Comme tous les produits venant directement des States Tomsrtbt va démarrer par défaut avec un clavier US. Ce point est particulièrement pénible. Il y a bien un écran qui permet de sélectionner son clavier dans une liste d'une quarantaine de claviers possibles, mais cet écran n'est maintenu que pendant 15 sec, après quoi la configuration par défaut se met en place. Et bien entendu, c'est précisément au moment où l'écran va apparaître qu'on vient vous déranger, que le téléphone sonne etc.. Il reste bien sûr la solution de rebooter mais ça fait vraiment utilisateur de ***** (un système d'exploitation né du côté de Redmond) . Une autre solution, bien plus évoluée, consiste à laisser booter jusqu'au bout puis à modifier le clavier au moyen de la commande loadkmap. A cet endroit il convient de noter que le shell de Tomsrtbt n'est (malheureusement ! ?) pas bash mais sh. Il s'en suit que la syntaxe des commandes prend une allure un peu inhabituelle qui fait largement appel à la redirection des flux. En clair, si vous vous êtes laissé "avalé" par le boot de Tomsrtbt il ne vous reste plus qu'à taper :

```
# loadkmap < /lib/key/fr.map
```

Compte tenu du fait que vous utilisez un clavier US à ce moment il faudra en fait taper :

```
# loqdk,qp < !lib!key!fr.,qp
```

en utilisant le point du pavé numérique ! Simple non ?

Entre nous il y a bien mieux à faire.

2.2 Fichiers de configuration

Pour trouver une solution plus performante il convient de chercher comment est enregistrée la configuration de Tomsrtbt. Le plus simple à cet endroit consiste à booter sur une disquette Tomsrtbt puis, une fois le système opérationnel, à 'monter' la disquette à la recherche des fichiers de configuration.

Pour ceci tapez :

```
# mount /dev/fd0u1722 /fl
```

2.3 Retour à la sélection du clavier

ce qui a pour effet de 'monter' la disquette dans le sous répertoire /fl de l'arborescence proposée par Tomsrtbt. Il faut en effet bien garder présent à l'esprit qu'au chargement TOMSRTBT transfère tout le contenu de la disquette en mémoire vive et qu'à la fin du boot, non seulement la disquette est inutile et peut à ce titre être retirée, mais de plus elle n'est même pas montée dans l'arborescence ! C'est donc à vous de faire ce montage. Une fois cette opération effectuée vous avez accès aux viscères de l'animal, et vous découvrez, à côté du fichier principal contenant les binaires, quelques fichiers dont on devine qu'ils jouent un rôle essentiel dans la configuration, rc.custom.gz et settings.s.

```
-rwxr-xr-x  1 505      100          4536 May 12  2001 boot.b
-rwxr-xr-x  1 root    root          829673 May  2  2002 bz2bzImage
-rwxr-xr-x  1 505      100           953 May  4  2002 settings.s
-rw-----  1 root    root         12800 May  4  2002 map
drwxr-xr-x  2 root    root           288 Sep 24 19:56 .
-rwxr-xr-x  1 root    root          1565 Nov  1 16:23 rc.custom.gz
drwxr-xr-x 14 root    root           544 Nov  1 22:07 ..
```

Le fichier qui nous intéresse ici plus particulièrement est rc.custom.gz. Il s'agit en fait d'un fichier script sh qui règle le démarrage de Tomsrtbt. La première étape va donc consister à le décompresser au moyen de gzip avec la syntaxe :

```
# gzip -d rc.custom.gz
```

laquelle va donner naissance à un nouveau fichier rc.custom. Je concède à cet endroit qu'il serait logique d'imaginer adapter le fichier à nos besoins avant de le recompresser. En pratique je ne me suis jamais donné la peine de recompresser, le fichier en format naturel tient sur la disquette sans problème. Attention, ce raisonnement est valable pour la version 2.0.103 de tomsrtbt parce que la disquette n'est occupée qu'à 97% ! Une version future dans laquelle la disquette serait pleine comme un oeuf pourrait conduire à une impossibilité de décompresser rc.custom.gz "sur place".

Concernant l'édition du fichier de configuration Tomsrtbt met à votre disposition VI, l'éditeur de base de tout système UNIX / LINUX. Nous sommes loin du confort de Emacs mais bon, nous sommes sur un système à une disquette !

2.3 Retour à la sélection du clavier

J'ai cherché longtemps la bonne solution pour résoudre mon problème. Bien sûr il y a moyen de placer, vers la fin du fichier, une ligne contenant

```
loadkmap < /lib/key/fr.map
```

ce qui aura pour effet de charger le clavier français après que le script aura fait son travail de paramétrage. A cet endroit il est possible aussi, pour éviter d'attendre les 15 secondes fixées par le script, de réduire ce temps à 1 sec, voir même 0 sec, ce qui rend presque invisible le questionnement par le script. Aucun de ces pis-allers n'étant intellectuellement satisfaisant il

m'est venu une autre idée basée sur le constat que Tomsrtbt n'a besoin que d'une bonne seconde entre la mise en place du clavier et l'invite d'identification de l'utilisateur, c'est à dire la fin du chargement. Pourquoi ne pas allonger les 15 sec dans des proportions sans rapport avec une utilisation normale de telle sorte que la question concernant le clavier reste posée pendant 5 min, 1/4 d'heure ou même une heure ? L'évènement perturbateur a le temps de s'éloigner après quoi vous avez toute latitude, dans le plus grand calme, de taper "fr" ou "16" pour demander à Tomsrtbt de charger la clavier français. Autre avantage, si vous travaillez dans un environnement international et que vous utilisez à l'occasion Tomsrtbt sur une machine qui utilise un clavier allemand ou autre, vous conservez la possibilité de charger l'un quelconque des claviers étrangers.

Pour faire la modification localisez les lignes à partir de la séquence "cat /lib/key.list". Cette ligne marque en pratique le début de la routine sh qui va déterminer la nature du clavier. Cette commande provoque l'affichage des choix de claviers. Deux lignes plus bas vous trouvez une indication (15 seconds) qui n'a plus lieu d'être et que vous pouvez donc effacer. Mais le vif du sujet est atteint 4 lignes plus bas encore avec la commande sleep 15. C'est ici que la temporisation est définie, c'est donc ici qu'il va falloir intervenir en remplaçant 15 par 3600 par exemple, si on souhaite une temporisation de 1 h. Tout ceci étant fait il suffit maintenant de sauvegarder rc.custom sur la disquette pour que la nouvelle configuration entre en action lors du prochain boot.

2.4 Paramétrage de la connexion réseau

Par défaut au lancement la carte réseau eth0 est configurée à l'adresse 1.1.1.1 ce qui est assez peu conventionnel. Les petits réseaux privés sont généralement des réseaux de classe C utilisant les adresses 192.168.1.0 à 192.168.255.255. Les adresses se terminant par "0" ou "255" ont généralement une signification particulière et ne sont pas affectés à des postes. A chaque lancement il faut donc, pour l'instant, utiliser ifconfig pour paramétrer correctement l'adresse de la carte. Une solution plus judicieuse consiste ici à choisir une adresse dans le haut de la plage, probablement toujours libre et à l'enregistrer dans les fichiers de configuration. Pour cela je propose la démarche suivante :

- créer deux variables IPADDR et NETMASK qui contiendront les adresses de la carte et masque de sous-réseau,
- configurer le démarrage de telle sorte que le paramétrage utilise ces deux variables.

L'examen des fichiers settings.s et rc.custom montre en effet que si le second est un script de lancement, le premier est un fichier de configuration définissant un certain nombre de variables comme le mot de passe administrateur et autres. En pratique il convient donc d'ajouter (par exemple) :

```
IPADDR=192.168.1.254
NETWORK=192.168.1.0
```

en tête du fichier settings.s. Aucune autre modification n'est nécessaire, le script rc.custom prévoit déjà l'utilisation des variables IPADDR et NETMASK.

Ces modifications configureront la carte eth0 à l'adresse 192.168.1.254 qui a peu de chance

d'être déjà occupée sur un petit réseau. En cas de problème il est de toutes façons toujours possible de refaire un petit ifconfig pour affecter une adresse libre.

3 Prise de contrôle à distance, telnet

3.1 Démon telnetd

Tomsrtbt est parfait pour reconnaître automatiquement les cartes réseaux les plus habituelles. Un petit coup de ifconfig (éventuellement) eét vous voilà sur le réseau à faire des pings de joie ! Seulement voilà, les pings ne servent qu'à vérifier que tout fonctionne, ils ne servent à rien d'autre. Pour aborder les activités sérieuses il faut pouvoir utiliser telnet et ftp. C'est par ces moyens qu'on peut prendre le contrôle à distance (telnet) de la machine ou transférer des fichiers (ftp) via le réseau.

Constat immédiat : en fouillant la disquette on trouve rapidement des fichiers nommés telnet et telnetd. On pourrait imaginer être sauvés. Que neni ! Telnetd se lance bien mais reste désespérément silencieux si on essaye d'entrer en dialogue avec lui depuis un autre poste du réseau :

```
[jml@jml3 jml]$ telnet jml1
Trying 192.168.1.254...
telnet: Unable to connect to remote host: No route to host
```

En fait telnetd ne doit pas être lancé directement mais doit être appelé par un logiciel qui surveille les flux sur les différents sockets, un peu comme le fait inetd ou xinetd sur un système LINUX de taille confortable. En fait c'est la lecture de rc.custom qui va nous mettre sur la voie. Dans les lignes de commentaires (qui commencent par un dièse), on trouve une ligne :

```
# c7:5:respawn:/usr/bin/nc -l -p 23 -e /usr/bin/telnetd
```

La lecture de l'aide en ligne (maigre) relative à la commande nc et de la documentation présente sur Internet, nc est l'abréviation de netcat, nous apprend que nc peut surveiller (-l = listen, écouter) un port particulier (-p 23, port dédié aux communications telnet) puis lancer telnetd en cas de besoin (-e exécute). La même étude nous apprend par ailleurs qu'il y a deux possibilités d'utiliser nc, soit en ligne de commande en tapant :

```
/usr/bin/nc -l -p 23 -e /usr/bin/telnetd
```

ou alors en intégrant la ligne dans le script de lancement.

La première solution permet surtout de faire des tests, elle n'est pas très pratique car elle ne permet qu'une connexion, nc se ferme aussitôt la connexion terminée. La solution est donc de faire lancer nc via le script de lancement lequel veillera, au moyen de respawn, à relancer un nouvel exemplaire de nc à chaque fermeture. Supprimez donc le caractère de commentaire qui marque le début de la ligne, ainsi que le dièse qui débute la ligne :

```
# kill -HUP 1
```

située deux lignes plus bas. Cette deuxième modification va demander au processus init (numéro 1) de relire le fichier de configuration inittab nouvellement modifié par ajout d'une ligne (voir cat»inittab etc.. quelques lignes plus haut).

3.2 Client telnet

L'utilisation du client telnet est tout à fait classique et n'appelle pas de commentaire supplémentaire.

4 Transfert ftp

4.1 Client ftp

Une recherche type “find | grep ftp” donne de suite la dimension du problème. Pas de binaire nommé ftp ou ftpd à l'horizon ! Et pourtant il doit bien y avoir un moyen de transmettre des fichiers via le réseau, sinon tout ce travail ne servirait à rien ! La lecture de la FAQ donne un premier élément de réponse, la commande wget serait le sésame pour les transferts http et ftp. Ouais, vite dit ! La (petite) page man de wget dans Tomsrftb nous en apprend un peu plus en nous donnant la syntaxe à appliquer :

```
wget [-c] [-O file] url
```

avec url ayant comme valeur

```
ftp://USER:PASSWORD@ftp.host.com:21/path/file.tgz
```

Pour bien expliquer je vais donner en exemple le détail du test que j'ai fait chez moi pour essayer le transfert ftp. Partant d'un réseau Ethernet sur lequel tournent quelques machines, la principale étant à l'adresse 192.168.1.3 (Mandrake 9.1), la machine de test Tomsrftb étant sur le même réseau, mon login sur la machine principale étant jml, mot de passe kronenbourg (ce qui n'est bien évidemment pas le reflet de la réalité) il vient :

```
wget -c -O test.tex ftp://jml:kronenbourg@192.168.1.3:21/home/jml/tomsjml.tex
```

Cette commande va transférer via ftp le fichier tomsjml.tex (source de l'article que vous lisez en ce moment) de /home/jml/ vers le répertoire courant de la machine tournant sous Tomsrftb en le renommant test.tex. Notez que j'ai fait fortuitement (une erreur de frappe) l'essai de ne pas préciser le numéro de port 21, le fichier a été transféré à l'identique. Je ne chercherais pas à analyser les raisons, gageons que cette approximation pourrait ne pas toujours être fonctionnelle.

4.2 Serveur ftp

4.2.1 Transfert de fichiers à l'unité

Les choses vont en se compliquant. Je n'ai pas trouvé de moyen simple de faire de Tomsrftb un serveur ftp au sens classique du terme. Par contre en fouillant un peu la syntaxe de nc "netcat", on peut trouver une méthode assez curieuse de transférer un fichier (ou un disque entier) d'une machine vers une autre via le réseau. En fait les deux machines doivent tourner toutes les deux sous Tomsrftb pour que l'opération soit simple. Bootez, mettez au point les adresses IP, vérifiez le tout avec le ping incontournable puis sur la machine de destination tapez :

```
# nc -l -p Port > fichier_destination
```

puis sur la machine source tapez

```
# dd if=fichier_à_copier | nc IP_dest Port
```

On pourra utiliser par exemple le port 21 normalement employé pour les transferts ftp. La première opération met la machine destinataire en écoute sur le port désigné puis transfère les données reçues vers un fichier du répertoire courant, la seconde envoie la sortie d'une commande de lecture dd au moyen d'un pipe vers netcat, et de fait vers la machine destination désignée par IP_dest. Il va sans dire que les numéros de ports devront être identiques.

4.2.2 Copie d'un disque complet

Contrairement au cas précédent il ne s'agit plus ici de récupérer un ou deux fichiers mais bien plutôt de transférer le contenu d'un disque complet. Cette opération pourra s'avérer nécessaire dans deux cas :

- Crash du système d'exploitation en place et tentative de sauver ce qui peut encore l'être avant de faire un formatage destructeur
- Sauvegarde d'un disque pour des raisons de sécurité, en vue d'une modification ou toute autre raison qui justifie qu'on veuille récupérer tout le contenu du disque.

L'opération se fera sur disque non monté, l'utilitaire de copie, dd, s'adressant directement au "device" et non plus aux fichiers individuels. On obtiendra sur l'autre machine un fichier qui sera en fait l'image du disque.

La syntaxe à mettre en oeuvre sera la suivante, tout d'abord sur la machine destination :

```
# nc -l -p Port > fichier_destination
```

puis sur la machine source :

```
# dd if=/dev/nom_du_device | nc IP_dest Port
```

A l'arrivée on obtiendra donc une image du disque source laquelle, pour être utilisable, devra être montée dans l'arborescence du PC de destination au moyen de la commande :

```
# mount fichier_destination -t type -o loop /mnt/test
```

ou type désigne le type de système de fichier, ext2, minix, vfat etc. en fonction du système de fichier du disque source. On suppose bien entendu que le point de montage /mnt/test est créé à l'avance et disponible.

Une possibilité intéressante pour limiter le volume du fichier qui transite sur le réseau est de comprimer les informations comme suit :

```
# dd if=/dev/nom_du_device | gzip -c | nc IP_dest Port
```

après quoi il faudra, après réception, décompresser le fichier reçu au moyen de `gzip -d` avant de le monter

5 Ajout de fonctionnalités supplémentaires

Tomsrtbt est capable de “monter” les disques locaux, de faire des copies de fichiers etc. Par contre en fouillant un peu il apparaît assez rapidement certaines limitations, parmi lesquelles le fait que Tomsrtbt ne voit pas les clés USB pourtant si pratiques quand il s’agit d’avoir toujours sur soi les fichiers les plus importants. Mais avant d’aborder cet aspect de la question faisons le pont sur les possibilités offertes par l’utilisation de lecteurs ZIP et de Tomsrtbt.

La visite du site de Tom OEHSER permet de découvrir assez vite des liens intéressants, suivre “wikiwiki” puis “other floppy things”. On arrive alors dans une espèce de caverne d’Ali-Baba contenant tout ce dont on peut rêver en pareille circonstance, notamment les modules USB ou ceux qui correspondent aux lecteurs ZIP les plus courants, 100 et 250 MO.

5.1 Connecter un lecteur ZIP

J’ai fait l’essai avec un lecteur classique fonctionnant sur port parallèle et utilisant des disquettes de 250 MO. Pour faire fonctionner ce matériel procéder comme suit :

- Récupérer le fichier `imm.o.bz2` sur le site décrit précédemment.
- Copier ce fichier sur une disquette.

Ce travail peut bien entendu être fait à partir de n’importe quel PC tournant sous LINUX, MAC OS X, voir même autre chose. Le format de la disquette importe peu, pour ma part je suis resté au format 82 secteurs, 21 pistes par secteur procurant 1722 ko. La ligne de commande pour “monter” une telle disquette est rappelée plus haut. Il faut ensuite :

- Démarrer la machine sous Tomsrtbt, se loguer correctement
- Remplacer la disquette de boot par la disquette comportant le fichier `imm.o.bz2`
- Monter cette disquette et se déplacer dans la nouvelle branche de l’arborescence.
- Décompresser avec `bzip2 -d imm.o.bz2`
- Charger le module avec `insmod imm.o`

Le lecteur doit répondre à cette commande par un petit bruit du mécanisme de tête de lecture. Une autre possibilité, bien plus simple, consiste à booter alors que le lecteur ZIP est déjà en place et sous tension. Tomsrtbt reconnaît alors automatiquement celui-ci sans autre forme de procès, la suite reste toutefois pertinente, même dans ce cas. Attention, j’ai fait le constat que ce raccourci fonctionnait parfaitement avec un lecteur 250 Mo utilisant `imm.o`, je ne peux pas

5.1 Connecter un lecteur ZIP

garantir que le fonctionnement sera aussi bon avec un lecteur 100 Mo utilisant le module ppa.o. C'est la raison pour laquelle j'ai conservé l'explication ci-dessus, valable également si le lecteur est ajouté alors que Tomsrtbt est déjà lancé. Il ne reste plus qu'à créer un sous-répertoire qui servira à recevoir le lien vers le contenu de la disquette ZIP avec

```
mkdir /zip
```

puis à "monter" la disquette ZIP avec :

```
mount -t vfat /dev/sda4 /zip
```

L'indication du type de fichier est indispensable faute de quoi Tomsrtbt va se plaindre de ne pas trouver de système de fichier ext3 sur /zip. Le fait qu'on utilise sda4 et non sda1 est lié à un souci de compatibilité avec un système d'exploitation édité du côté de Redmond. Tomsrtbt étant une distribution de dépannage plus qu'une distribution de travail il est donc important de conserver la compatibilité avec certains systèmes même si ceux-ci ne font pas partie de la grande famille UNIX.

Attention, les essais ont été réalisés avec un lecteur 250 Mo. Je n'ai pas vérifié si un lecteur 100 Mo serait reconnu automatiquement au boot. En cas de configuration manuelle utiliser le module ppa et non imm pour ce type de lecteur.