

GIMP et scanner sous FreeBSD

Jean-Marc LICHTLE

22 mai 2005

Table des matières

1 Introduction	1
2 Compilation de GIMP et de XSANE	1
3 Configuration de FreeBSD	2

1 Introduction

J'ai décidé de traiter dans le même document de deux sujets connexes, l'installation de GIMP et de SANE et le paramétrage de l'ensemble pour que l'utilisateur lambda puisse utiliser l'acquisition d'image directement via le menu File / Acquire / XSane : Device dialog.... Ce document décrit une installation FreeBSD 5.4 et un scanner raccordé sur port USB.

2 Compilation de GIMP et de XSANE

On admettra que les ports sont installés et mis à jour correctement. Dans ces conditions la compilation des deux logiciels ne pose guère de problème :

- Établir la connexion avec Internet (l'ADSL ou toute autre forme de réseau rapide est bien entendu souhaitable).
- Se rendre dans /usr/ports/graphics/gimp/ et taper "make install clean" ce qui va provoquer le téléchargement des sources et la compilation de GIMP.
- De la même façon se rendre dans /usr/ports/xsane, mais cette fois taper "make WITH_GIMP = yes" pour lancer la compilation puis make install clean pour terminer la procédure.

Le "make WITH_GIMP=yes va configurer XSANE pour qu'il puisse être utilisé comme plug-in par GIMP. Les deux logiciels étant compilés et installés vous pouvez maintenant passer au tout dernier détail, créer un lien symbolique de GIMP vers XSANE (le fameux plug-in évoqué plus haut) en tapant, dans votre répertoire home :

```
$ ln -s /usr/X11R6/bin/xsane .gimp-2-2/plug-ins/xsane
```

Bien entendu il faudra veiller à adapter la syntaxe, celle-ci étant correcte pour la version 2.2 de gimp, pas les autres (évidemment).

Arrivé à ce stade vous devriez pouvoir vérifier que :

- GIMP a bien pris en compte la présence de XSANE, voir menu File / Acquire, la ligne XSane... qui vient d'apparaître.
- Que l'acquisition via XSane ne fonctionne toutefois pas (message d'erreur indiquant qu'aucun scanner n'a été trouvé).
- Que, par contre, à condition de créer un lien semblable pour le compte root, ce dernier arrive bien à accéder à XSane, malgré quelques injures du système qui rappelle, fort justement, que l'emploi de XSANE par root ne fait pas précisément partie des tâches d'administration normales.

3 Configuration de FreeBSD

Le problème vient tout simplement du fait que le périphérique (ils disent le "device") /dev/usb/lcd0 est réservé à l'usage de root ou d'un utilisateur faisant partie du groupe "operator". Les droits sur ce périphérique sont fixés à 660 c'est à dire que le pecus vulgaris qui souhaite se servir du scanner se fait proprement jeter à cet endroit.

Une première solution serait d'abonner l'utilisateur au groupe operator qui est autorisé à accéder au scanner. Pour ma part j'ai choisi une autre technique, peut-être moins académique, qui consiste à ouvrir les droits à tous les utilisateurs en les fixant à 666. De cette façon tout utilisateur du PC pourra accéder au scanner sans restriction ce qui ne me semble pas une incongruité du point de vue sécurité. Un petit coup de :

```
# chmod 666 /dev/usb/lcd0
```

suivi d'un

```
# chmod 666 /dev/usb/lcd0
```

et l'affaire est dans le sac, vous pouvez désormais accéder à votre scanner depuis GIMP. Joie de courte durée puisqu'il suffit de rebooter pour se retrouver dans la situation précédente. Sous FreeBSD Le répertoire /dev n'est en effet pas un sous répertoire "physique" mais une construction logée en RAM et refaite à chaque boot. L'enjeu est donc de trouver le moyen d'enregistrer définitivement cette modification.

En fait le comportement de devd, le démon qui assure la gestion des périphériques, est dicté par deux fichiers, /etc/devd.conf et /etc/devfs.conf. C'est au niveau de ce dernier qu'il va falloir intervenir pour ajouter les lignes :

```
perm usb/lcd0 0666
perm usb/lcd0 0666
```

Ces deux lignes vont imposer un niveau d'autorisation égal à 666 sur les deux périphériques /dev/usb0 et /dev/uscanner0. Ce faisant à chaque boot les droits seront configurés correctement et tout utilisateur autorisé sur la machine pourra accéder au scanner.

jml