

# Linux ADSM Mini-Howto

---

par Thomas König, [Thomas.Koenig@ciw.uni-karlsruhe.de](mailto:Thomas.Koenig@ciw.uni-karlsruhe.de)

v, 15 Janvier 1997

Ce document décrit l'installation et l'utilisation sur un poste de travail Linux d'un client pour le système de sauvegarde commercial ADSM.

## Contents

<a href="#">1 Introduction</a>	1
<a href="#">2 Installation du module iBCS</a>	1
<a href="#">3 Installation du client ADSM</a>	2
<a href="#">4 Fonctionnement du client</a>	3
<a href="#">5 Problèmes identifiés</a>	3

## 1 Introduction

ADSM est un système de sauvegarde en réseau vendu par IBM que l'on retrouve dans de nombreuses organisations. Les programmes clients existent pour une large palette de systèmes : différents Unix, Windows, Novell, Mac, Windows NT. Malheureusement, à la date de rédaction de ce document, aucune version spécifique à Linux n'existe.

L'utilisation d'ADSM implique donc le recours au binaire SCO et à l'émulateur iBCS2. Il est ici question d'ADSM v2r1.

A ce jour, je ne connais qu'une version fonctionnant sous Linux, et uniquement sur plate-forme i386.

## 2 Installation du module iBCS

Le module iBCS est disponible à l'adresse suivante : <ftp://tsx-11.mit.edu/pub/linux/BETA/ibcs2> . [NdT : un miroir français <ftp://ftp.lip6.fr/pub/linux/tsx-11/BETA/ibcs2> ] Si vous employez une version 1.2.13 du noyau, récupérez [ibcs-1.2-950721.tar.gz](#), décompactez-le et appliquez les patches [ibcs-1.2-950808.patch1](#) et [ibcs-1.2-950828.patch2](#). Vous pouvez alors invoquer "make" et insérer le module iBCS via "insmod".

Pour un noyau 2.0, récupérez [ibcs-2.0-960610.tar.gz](#), décompressez-le dans un répertoire adéquat, allez dans ce répertoire et appliquez le patch ci-dessous :

```
--- iBCSemul/ipc.c.orig Wed Jan 15 21:32:15 1997
+++ iBCSemul/ipc.c      Wed Jan 15 21:32:31 1997
@@ -212,7 +212,7 @@
     switch (command) {
         case U_SEMCTL:
```

```

        cmd = ibcs_sem_trans(arg3);
-       arg4 = (union semun *)get_syscall_parameter (regs, 4);
+       arg4 = (union semun *)(((unsigned long *) regs->esp) + (5));
        is_p = (struct ibcs_semid_ds *)get_fs_long(arg4->buf);

#ifdef IBCS_TRACE
        if ((ibcs_trace & TRACE_API) || ibcs_func_p->trace)

```

Copiez ensuite `CONFIG.i386` en `CONFIG` et invoquez `make`.

S'ils ne sont pas déjà présents, créez les périphériques requis en exécutant :

```

# cd /dev
# ln -s null XOR
# ln -s null XOR
# mknod socksys c 30 0
# mknod spx c 30 1

```

### 3 Installation du client ADSM

Le binaire `SCO` est divisé en trois fichiers `tar`. Positionnez-vous à la racine de votre système de fichiers, vérifiez la valeur de `umask` et décompactez-les en tant qu'utilisateur `root`. Un script d'installation apparaîtra dans votre répertoire `/tmp`. Invoquez-le.

Éditez ensuite `/usr/adsm/dsm.sys` et `/usr/adsm/dsm.opt`. Certaines lignes du fichier `dsm.sys` requièrent une attention particulière :

#### Servername

le nom du serveur

#### TCPServeraddress

le nom complet du serveur (FQDN)

#### NODename

le nom de votre station

Vous préciserez dans le fichier `dsm.opt` :

#### Server

comme précédemment

#### Followsymbolic

s'il faut suivre les liens symboliques (c'est rarement une bonne idée)

#### SUmdir

faut-il sauvegarder les sous-répertoires ? (généralement oui)

#### domain

le système de fichiers à archiver

Créez ensuite un fichier `/etc/mnttab` à la mode `SCO` à partir de votre `/etc/fstab`. Le script Perl suivant, `fstab2mnttab`, le fera pour vous.

---

```
#!/usr/bin/perl

$mnttab_struct = "a32 a32 I L";

open(MTAB, "/etc/mtab") || die "Impossible d'ouvrir /etc/mtab: $!\n";
open(MNTTAB, ">/etc/mnttab") || die "Impossible d'ouvrir /etc/mnttab: $!\n";

while(<MTAB>) {
    next if /pid/;
    chop;
    /^(\S*)\s(\S*)\s(\S*)\s.*$/;
    $device = $1;
    $mountpt = $2;
    $fstype = $3;
    if($fstype ne "nfs" && $fstype ne "proc") {
        $mnttab_rec =
            pack($mnttab_struct, $device, $mountpt, 0x9d2f, time());
        syswrite(MNTTAB, $mnttab_rec, 72);
        print "Entrée créée pour : $device $mountpt $fstype\n";
    }
}

close(MNTTAB);
exit 0;
```

---

Le client ne réclame aucune bibliothèque dynamique : il est lié statiquement.

## 4 Fonctionnement du client

Il existe deux clients. `dsm` propose une interface X11 tandis que `dsmc` reste en ligne de commande. Votre service informatique vous en dira davantage à leur sujet. Un script de démarrage du style :

```
dsmc schedule -quiet 2>&1 >/dev/null &
```

s'avèrera vraisemblablement nécessaire.

## 5 Problèmes identifiés

SCO se révèle malheureusement incapable de supporter les noms de machines dépassant les 8 caractères. Si le nom de votre machine dépasse cette limite ou qu'il inclut le nom de domaine, vous devrez le préciser dans la ligne `NODename` du `/usr/adsm/dsm.sys`.

Le recours à la variable `DISPLAY` requiert de la part de celle-ci qu'elle comprenne le nom complet de votre machine, c'est à dire `DISPLAY=maMachine.monDomaine:0` au lieu de `DISPLAY=maMachine:0`.