

BOOT

Ms-Dos; Windows 95/98; Windows NT

*** * * * SOMMAIRE * * * ***

Structure d'un disque et de ses partitions (rappels)

Structure du secteur de partition (rappels)

Structure d'une entrée dans la table de partition (rappels)

Structure d'un secteur de boot (rappels)

Mécanisme de boot d'un PC (*DOS; Windows 95/98; Windows NT*)

Démarrage de Windows 95/98 sans exécuter les fichiers de démarrage

Sauvegarde et restauration du secteur de boot

Triple boot DOS (6.x) - Windows 95/98 - Windows NT

En cas de malheur avec Windows 95 (98)...

En cas de malheur avec Windows NT...

Comment installer DOS (6.22) après Windows 95 (98)

Les autres outils...

Le fichier MSDOS.SYS (Windows 95 /98)

Raccourcis avec la touche "Windows" (claviers 105 touches)

Configuration du panneau de configuration (Windows95/98)

Structure et paramètres du fichier "boot.ini" (Windows NT)

Structure et paramètres du fichier "boot.ini" (Windows NT)

Création d'une disquette de boot partiel sous Windows NT

Les séquences du démarrage de Windows NT

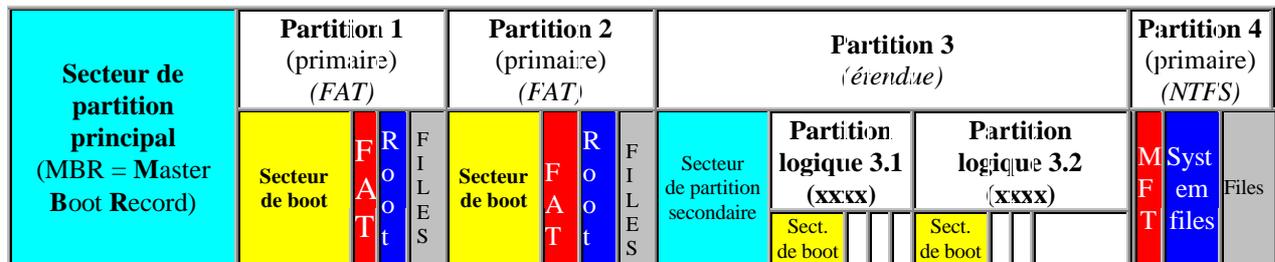


Structure d'un disque et de ses partitions

Un disque dur peut être partagé en plusieurs "morceaux", appelés "**partitions**", pour différentes raisons, par exemple:

- Coexistence de différents systèmes d'exploitation (DOS, Windows NT, LINUX, OS/2,...)
- Taille physique d'un disque trop grande par rapport à ce que peut gérer le système d'exploitation utilisé (p.ex. les premiers DOS ne pouvaient pas "adresser" des partitions de plus de 32 M octets, les suivants plus de 2 G octets,...)
- Séparation physique, pour des questions de sécurité, du système, des applications, des données,...

Un disque (physique) est donc ainsi structuré (*exemple*) :



FAT : *File Allocation Table*, a donné son nom au type de partition du même nom, utilisé par DOS, Windows 95/98, Windows NT

la FAT est une table de chaînage de clusters (agrégats de secteurs) permettant l'adressage et la reconstitution des fichiers

NTFS : *NT File System*, utilisé par Windows NT

MFT : *Master File Table*, un composant de NTFS

L'organisation d'un disque est créée par un utilitaire, qui s'appelle, par exemple sous *DOS* ou *LINUX*, **FDISK**. Le secteur de partition est situé au début du disque (*tête 0, cylindre 0, secteur 1*), comme tout secteur il a une taille de **512** octets.

Il est chargé par le **BIOS** à l'adresse mémoire **0000:7C00** (si une disquette n'a pas "pris la main").

Ensuite, le BIOS teste si le **dernier** mot (de 16 bits) de ce secteur est égal à **0xAA55**, qui indique que le disque a bien été partitionné (sinon, un message d'erreur est affiché par le BIOS). Si c'est effectivement le cas, le **programme de partition** qui réside au début de ce secteur est lancé.

Structure du secteur de partition

Adresse (hexa)	Contenu	Type
000	Programme de partition	Code (446 octets au maximum)
1BE	1ère entrée dans la table de partition	16 octets
1CE	2ème entrée dans la table de partition	16 octets
1DE	3ème entrée dans la table de partition	16 octets
1EE	4ème entrée dans la table de partition	16 octets
1FE	AA55 (code d'identification)	2 octets

On constate qu'il **ne peut pas** y avoir **plus** de **4** partitions dites **primaires** (cette limite peut être contournée à l'aide des partitions **étendues**)

Structure d'une entrée dans la table de partition

Adresse (hexa)	Contenu	Type
00	Etat de la partition : <ul style="list-style-type: none"> • 00 : partition non active • 80 : partition active 	1 octet
01	Tête où commence la partition	1 octet
02	Secteur et cylindre où commence la partition	2 octets
04	Type de partition : <ul style="list-style-type: none"> • 00 : non utilisée • 01 : FAT12 (< 10 M) • 02 : XENIX • 03 : XENIX • 04 : FAT16 (< 32 M) • 05 : FAT, HPS, NTFS étendue • 06 : FAT16 (> 32 M et <2 G) • 07 : HPFS/NTFS • ... • 0B : FAT32 (>2 G et <2047 G) • 0C : idem 0B + LBA • 0E : idem 06 + LBA • 0F : idem 05 + LBA • • 81 : LINUX • 82 : LINUX • 83 : LINUX (Swap) • 	1 octet
	LBA = <i>Large Block Addressing</i> , dispositif du BIOS permettant l'accès à des disques de plus de 1024 cylindres	
05	Tête où finit la partition	1 octet
06	Secteur et cylindre où finit la partition	2 octets
08	Distance en secteurs entre secteur de partition et secteur de boot de la partition	4 octets
0C	Nombre de secteurs de la partition	4 octets

Une partition **étendue** se compose d'une **nouvelle table** de partition, de **structure identique** à celle de la table principale, permettant ainsi un **chaînage** vers des partitions supplémentaires "encapsulées" à la suite.

Structure d'un secteur de boot

Cette structure varie suivant le système d'exploitation. Ci-dessous est décrite celle des **partitions FAT**

Adresse (hexa)	Contenu	Type
000	Instruction de saut à la routine de boot	3 octets
003	Nom du fabricant et n° version (p.ex. MSWIN4.0)	8 octets
00B	Octets par secteur (0x200 = 512)	2 octets
00D	Secteurs par cluster (variable)	1 octet
00E	Nbr secteurs réservés (1)	2 octets
010	Nbr de FAT (2)	1 octet
011	Nbr d'entrées répertoire racine (0x200 = 512)	2 octets
013	Nbr de secteurs dans le volume (variable)	2 octets
015	Descripteur de support (0xF8 pour un disque dur)	1 octet
016	Nbr de secteurs par FAT	2 octets
018	Nbr de secteurs par piste (variable)	2 octets
01A	Nbr de têtes (variable)	2 octets
01C	Distance entre 1er secteur du volume et 1er secteur du support	2 octets
01E à 1FD	Routine de boot	482 octets
1FE	AA55 (code d'identification, id. secteur de partition)	2 octets

QUE SE PASSE T'IL A LA MISE SOUS TENSION

Mise /Tension ou Reset
↓
Bootstrap ROM FFFF0h chargé
Auto-test matériel

- Visu du prompt écran ?
- Affichage version BIOS ou info carte vidéo
- Test de la RAM (0 à x Ko) ?
- Voyants clavier ?
- Disque dur led +déplacement tête ?
- Disquette led + déplacement tête A: B: ?
- Reset imprimante ?
- Bip ? si oui combien ? ---

CTRL+ALT+SUPP

FAUTES DETECTEES ?

oui

Edition code erreur

Exemple clavier
20 301

STOP

non

non

TEST PRESENCE
DISK A: ?
DISK C: ?

Message d'erreur

STOP

A:

C:

Loader ROM
→ Loader OS
(Operating System)

Lecture et exécution du
MBR
(Master Boot Record)

Chargement en RAM du
secteur BOOT de A: et
exécution

Chargement en RAM du
secteur BOOT de la
partition active

Si fichiers système endommagés
affichage message d'erreur
NON SYSTEM DISK

Chargement des fichiers
systèmes
MS-DOS.SYS & IO.SYS

Si pas système affichage message d'erreur
NON SYSTEM DISK
(stop)

Si ligne de commande erronée ou fichier
absent: Affichage message d'erreur.
Insérer des remarques pour invalider certaines
commandes pour la mise au point.

Lecture fichier
CONFIG.SYS (root)
et exécution

- PRECAUTIONS A PRENDRE AVANT INTERVENTION**
1. Renommer les fichiers de configuration avant toutes modifications
 2. S'assurer qu'un BACKUP ait été effectué par le client
 3. Visualiser l'environnement avec SET
 4. Etre attentif à tous les messages visuels et auditifs

Si command .com absent ou mauvais affichage
d'un message d'erreur. BAD OR MISSING
COMMAND INTERPRETER

Chargement de
COMMAND.COM
(Root ou chemin du Comspec)

Visualisation rapide des commandes
Keyb; path; ...
Si fichiers introuvable affichage message
d'erreur

Lecture et exécution de
AUTOEXEC.BAT

Si pas AUTOEXEC.BAT
TIME ET DATE

Si OK
A: \> ou C: \>

Affichage du
prompt A: C :
et attente de
commande clavier

Mécanisme de boot d'un PC (DOS; Windows 95/98; Windows NT) 3/3

Le programme de partition, décrit précédemment, :

1. identifie la **partition active**, (définie soit par *FDISK*, par le *gestionnaire de disque* de Windows NT, ou par un outil comme *System Commander*)
2. détecte le **système d'exploitation** à lancer
3. **charge** le secteur de **boot** correspondant
4. **lance** le programme de **boot** qu'il contient

Ce (mini) programme cherche sur le disque un (ou deux) **autre(s) programme(s)** et lui passe la main.

Le nom de ce programme est également stocké dans le secteur de boot (à la fin généralement).

Ce(s) programme(s) s'appelle (nt) :

- pour **DOS** : *IO.SYS* et *MSDOS.SYS* (ou *IBM....COM*)
- pour **Windows 95/98** : *IO.SYS* (*MSDOS.SYS* n'étant plus qu'un fichier texte de config) + *WINBOOT.SYS*
NB: ce dernier fichier est en réalité IO.SYS de Windows 95, renommé ainsi lors d'un démarrage du précédent DOS
- pour **Windows NT** : *NTLDR* ("NT" Loader)
- pour **LINUX** : *LILO* (=Linux Loader)

Dans le cas de DOS, il contient la majeure partie de l'OS. Dans le cas de NT (et aussi de Linux), ce programme n'est qu'une étape intermédiaire, avant de charger réellement l'OS.

- **Boot avec Windows NT installé :**
le mécanisme complet de démarrage est décrit ailleurs ("Les séquences du démarrage de Windows NT")
- **Boot avec DOS ou Windows 95/98 seul installé :**
Il y a exécution de *IO.SYS* et *MSDOS.SYS* **Boot avec DOS et Windows 95/98 installés :**
Dans le cas de **Windows 95/98**, si on a demandé de garder l'ancien système (DOS), la racine contient les fichiers supplémentaires :
 - *MSDOS.DOS*
 - *IO.DOS*
 - *COMMAND.DOS*
 - *AUTOEXEC.DOS*
 - *CONFIG.DOS*

Ce sont les 2 fichiers systèmes du **précédent** DOS (p.ex. 6.22), le processeur de commande et les 2 fichiers de démarrage habituels *AUTOEXEC.BAT* et *CONFIG.SYS*, tous renommés car Windows 95 utilise les mêmes. Quand on appuie sur la touche **F8** lors du démarrage de Windows 95, on peut choisir de booter sur l'ancien OS (DOS). Dans ce cas, Windows 95 effectue les opérations suivantes :

1) Renommage des fichiers Windows95		2) Renommage des fichiers DOS	
<i>MSDOS.SYS</i>	<i>MSDOS.W40</i>	<i>MSDOS.DOS</i>	<i>MSDOS.SYS</i>
<i>IO.SYS</i>	<i>WINBOOT.SYS</i>	<i>IO.DOS</i>	<i>IO.SYS</i>
<i>COMMAND.COM</i>	<i>COMMAND.W40</i>	<i>COMMAND.DOS</i>	<i>COMMAND.COM</i>
<i>AUTOEXEC.BAT</i>	<i>AUTOEXEC.W40</i>	<i>AUTOEXEC.DOS</i>	<i>AUTOEXEC.BAT</i>
<i>CONFIG.SYS</i>	<i>CONFIG.W40</i>	<i>CONFIG.DOS</i>	<i>CONFIG.SYS</i>

Le contrôle est passé alors à *IO.SYS* "DOS", mais le secteur de boot n'est pas modifié. Ainsi, si on reboote (donc sous DOS à ce moment là), le programme de boot va tester l'existence du fichier *WINBOOT.SYS*, et c'est lui qui va être chargé **à la place** de *IO.SYS*. Le renommage des fichiers va alors être inversé (*MSDOS.SYS* *MSDOS.DOS*,)

NB : Si on a renommé ou détruit sous DOS le fichier *winboot.sys*, Windows 95 **ne pourra pas démarrer**



Démarrage de Windows 95/98 sans exécuter les fichiers de démarrage

S'il y a lieu, vous pouvez démarrer l'ordinateur sans charger certains composants. Cette possibilité s'avère particulièrement utile lorsque vous rencontrez des problèmes en démarrant Windows 95/98.

Pour démarrer l'ordinateur en mode dépannage

1. Démarrez ou redémarrez l'ordinateur. Après le démarrage de celui-ci, Windows affiche le texte suivant:

Démarrage de Windows 95

2. Pendant que ce message est affiché, appuyez sur la touche F8 et relâchez-la. Un menu apparaît.

3. Tapez le numéro de l'option voulue, ou utilisez la touche BAS pour sélectionner, puis appuyez sur *ENTREE*.

Menu de démarrage de Microsoft Windows 95

1. *Normal*
2. *Journal de démarrage (\BOOTLOG.TXT)*
3. *Mode sans échec*
4. *Mode sans échec avec un support réseau*
5. *Confirmation pas à pas*
6. *Ligne de commande uniquement*
7. *Invite MS-DOS en mode sans échec uniquement*
8. *Version précédente de MS-DOS (si existe)*

Entrez un choix:

F5= Mode sans échec Maj+F5= Invite MS-DOS Maj+F8= Confirmer chaque ligne [N]

Le tableau suivant explique les différents choix qui vous sont offerts lorsque vous appuyez sur F8 durant le démarrage

Option de démarrage	EFFET
<i>Normal</i>	Démarre Windows normalement
<i>Journal de démarrage (\BOOTLOG.TXT)</i>	Démarre Windows normalement, mais crée un fichier texte appelé BOOTLOG.TXT dans le répertoire racine. Ce fichier contient des informations sur les fichiers qui ont été chargés correctement.
<i>Mode sans échec</i>	Démarre Windows avec une configuration de base à la place de la configuration habituelle (moniteur VGA, pilote souris Microsoft...). De ce fait, certaines parties du système risquent de ne pas fonctionner comme à l'accoutumée. Après le démarrage de Windows, vous pouvez faire des modifications et redémarrer l'ordinateur. Ce type de démarrage ne vous donne accès ni au CD-ROM ni aux imprimantes, ni aux autres périphériques !
<i>Mode sans échec avec support réseau</i>	Démarre Windows avec une configuration de base, mais inclut une prise en charge du réseau.
<i>Confirmation pas à pas</i>	Démarre l'ordinateur en affichant un message de confirmation à chaque étape du démarrage: pour exécuter la commande courante, appuyez sur <i>ENTREE</i> ; pour sauter cette commande, appuyez sur <i>ECHAP</i> . Cette option vous permet de charger seulement les composants que vous voulez.
<i>Ligne de commande uniquement</i>	Démarre l'ordinateur normalement, mais ne démarre pas l'interface Windows. Vous pouvez démarrer Windows en tapant <i>WIN</i> .
<i>Ligne de commande en mode sans échec</i>	Démarre l'ordinateur avec une configuration de base à la place de la configuration habituelle et ne démarre pas l'interface Windows
<i>Version précédente de MS-DOS</i>	Démarre MS-DOS en utilisant les fichiers qui ont été sauvegardés lors de l'installation de Windows. Cette version de MS-DOS est donc celle que vous exécutez avant d'installer Windows 95

Sauvegarde et restauration du secteur de boot

Le **secteur de boot** sert essentiellement à lancer

- soit le fichier principal du **système d'exploitation** (DOS, Windows 95),
- soit un **programme de chargement** ("*loader*") qui va permettre de choisir un système d'exploitation (NT, Linux, OS/2,...)

Ce secteur est donc **primordial**, et il peut être utile de le restaurer (donc il doit avoir été sauvegardé au préalable) dans le cas par exemple d'erreurs de manipulation, ou si l'on désire changer de système d'exploitation sans devoir tout réinstaller.

La **méthode** décrite ci-dessous fonctionne aussi bien pour *DOS, Windows 95/98* que pour *NT*.

Elle ne nécessite **aucun outil particulier**, autres que ceux fournis en standard (essentiellement "**debug**").

Dans tous les cas elle doit être effectuée :

- sous **DOS** ou dans une **fenêtre** de commande (sous Windows) en ce qui concerne la **sauvegarde**
- sous **DOS exclusivement** (à partir d'une disquette DOS p.ex., ou en "mode MS-DOS" sous Windows 95) en ce qui concerne la **restauration**

Pour cela, *éditer* les fichiers **batchs** suivants :

Fichier saveboot.bat	Fichier restboot.bat
<pre>@echo off if %1.==. goto syntaxe echo Sauvegarde du secteur de boot dans%1 c: cd \ REM creation du fichier de commandes REM debug echo l 100 2 0 1 >cmddbg.txt echo n %1 >>cmddbg.txt echo r cx >>cmddbg.txt echo 200 >>cmddbg.txt echo w >>cmddbg.txt echo q >>cmddbg.txt a:debug <cmddbg.txt echo Sauvegarde effectuee goto fin :syntaxe echo syntaxe : saveboot nom_de_fichier echo p.ex. : saveboot bootsect.622 :fin</pre>	<pre>@echo off if %1.==. goto syntaxe if NOT EXIST %1 goto erreur echo Restauration du secteur de boot echo depuis %1 pause c: cd \ REM creation du fichier de commandes REM debug echo n %1 >cmddbg.txt echo l >>cmddbg.txt echo w 100 2 0 1 >>cmddbg.txt echo q >>cmddbg.txt a:debug <cmddbg.txt echo Restauration effectuee goto fin :erreur echo **le fichier %1 n'existe pas ** goto fin :syntaxe echo syntaxe : restboot nom_de_fichier echo p.ex. : restboot bootsect.622 :fin</pre>

Explication : tout réside dans la ligne de commande de **debug** "l 100 2 0 1" (ou "w 100 2 0 1"), qui signifie "*lecture*" (ou "*écriture*") à l'adresse 100 du disque C (n°2), secteur n°0, 1 secteur (correspondant au secteur de boot du disque C)

Triple boot DOS (6.x) - Windows 95/98 - Windows NT

Le **but** à atteindre est d'obtenir un **menu de boot unique** (généré par NT) permettant le choix **direct** (sans devoir appuyer sur une touche de fonction telle que F4, F8, ...) entre :

- **DOS** (6.22 p.ex.)
- **Windows 95** (98)
- **Windows NT** (3.5x, 4, 5)

En effet, si l'on procède à l'installation des différents systèmes dans l'ordre (logique) suivant :

1. Installation de DOS
2. Installation de Windows 95 (en demandant de conserver la précédente version d'OS)
3. Installation de Windows NT (en demandant de conserver la précédente version d'OS)

Le boot de la machine se fera **au départ** sous NT, avec menu d'options NT / Windows 95. Si on désire alors accéder au DOS initial, il faudra **d'abord** choisir Windows 95, **puis** appuyer sur F8, pour enfin sélectionner "Précédente version...". ce qui n'est pas très contraignant, mais n'est pas très pratique non plus !

On préférerait certainement avoir un **menu unique**, lors du démarrage de NT.

Cela est rendu possible si le fichier "**boot.ini**" contient les 2 lignes suivantes :

```
.....
c:\bootsect.622="L'ancetre (Dos 6.22)" /win95dos
c:\bootsect.w95="La glute a Billou" /win95
multi(0).....
.....
```

Les 2 fichiers :

- c:\bootsect.622 et
- c:\bootsect.w95

sont les images des secteurs de boot, respectivement sous **DOS** et sous **Windows 95**. Leurs noms sont **arbitraires**.

Les commutateurs /win95dos et /win95 sont à préciser pour que **NT** émule le processus de multi-boot de Windows 95

Pour obtenir (ou recréer) les fichiers secteurs de boot :

utiliser [saveboot.bat](#) décrit précédemment.

Mais pour que tout cela "fonctionne", il faut que dans la racine du disque **C:** figurent les fichiers suivants :

<i>Fichiers DOS renommés</i>	<i>Fichiers Windows 95/98</i>
- MSDOS.DOS	- MSDOS.SYS
- IO.DOS	- IO.SYS
- COMMAND.DOS	- COMMAND.COM
- AUTOEXEC.DOS	- AUTOEXEC.BAT
- CONFIG.DOS	- CONFIG.SYS

Si les fichiers "DOS" n'existent pas (détruits, ou simplement parce **DOS** n'a jamais été installé):

1. **Rebooter** le PC avec une disquette **DOS 6.22** (par exemple)
2. Si les fichiers Windows **95 existent**, les renommer car la phase suivante va les écraser.
3. **Exécuter** la commande **a:sys c:** qui va transférer **IO.SYS**, **MSDOS.SYS** et **COMMAND.COM** de **DOS** sur le disque
4. **Installer** (éventuellement) les autres fichiers du DOS (*debug*, en particulier, est **obligatoire** pour la phase suivante)
5. **Restaurer** le secteur de **boot** de NT (qui vient d'être écrasé par la commande **sys** de DOS) par la commande :

```
c:\>restboot.bat bootsect.nt4
=> restauration du secteur de boot NT
```

Si les fichiers "Windows 95" n'existent pas (détruits) (NB: si Windows 95 n'a jamais été installé, procéder à son installation) :

1. **Rebooter** le PC avec une disquette **DOS formatée avec Windows 95**
2. Si les fichiers **DOS existent**, les renommer car la phase suivante va les écraser.
3. **Exécuter** la commande **a:sys c:** qui va transférer *IO.SYS*, *MSDOS.SYS* et *COMMAND.COM* de **Windows 95** sur le disque
4. **Installer** (éventuellement) les autres fichiers du répertoire *COMMAND* de Windows 95 (*debug*, en particulier, est **obligatoire** pour la phase suivante)
5. **Restaurer** le secteur de **boot** de NT (qui vient d'être écrasé par la commande **sys** de Windows 95) par la commande :

```
c:\>restboot.bat bootsect.nt4  
=> restauration du secteur de boot NT
```

Remarque : l'installation de **Windows 98** n'efface pas le secteur de boot de **NT**, se limitant à modifier le fichier **boot.ini** en ajoutant la ligne **c:\="Microsoft Windows 98"**

Dans tous les cas, éditer le fichier **boot.ini** de façon à insérer les 2 lignes relatives à DOS et à Windows 95. On peut alors rebooter la machine, et choisir directement entre **NT**, **Windows 95(98)** ou **DOS**.



En cas de malheur avec Windows 95 (98)...

Windows 95 ne redémarre plus, mais c'est DOS (ou un message d'erreur) que l'on voit à la place.

Vraisemblablement le secteur de boot de Windows 95 a été écrasé (p.ex. par celui de DOS) et/ou les fichiers systèmes de Windows 95 ont été détruits ou corrompus.

1. **Rebooter** la machine (si nécessaire) afin d'être sous **DOS** (depuis le disque dur ou depuis une disquette **système formatée** depuis **Windows 95**)
2. **Taper** la commande suivante :

```
a:\>sys c:  
restauration du secteur de boot 95  
recopie des fichiers système de 95 :  
- IO.SYS, MSDOS.SYS, COMMAND.COM
```

3. Si **NT** est **installé**, et que l'on désire encore l'utiliser, il faut **restaurer** le secteur de **boot** de **NT** (car il vient d'être écrasé par la commande **sys** de Windows 95). Pour cela, exécuter **sous DOS** la commande suivante :

```
c:\>restboot.bat bootsect.nt4  
=> restauration du secteur de boot NT
```

(La suite du lancement de 95 est évidemment subordonnée à la présence des fichiers de 95 dans le répertoire désigné dans le fichier `msdos.sys` : existence du répertoire `\windows`, `\windows\system`, ...)

En cas de malheur avec Windows NT...

NT ne redémarre plus, mais c'est DOS (ou Windows 95) que l'on voit à la place.

Vraisemblablement le secteur de boot de NT a été écrasé (p.ex. par celui de DOS ou celui de Windows 95).

1. **Rebooter** la machine (si nécessaire) afin d'être sous **DOS** (depuis le disque dur ou depuis une disquette)
2. **Taper** la commande suivante :

```
c:\>restboot.bat bootsect.nt4  
=> restauration du secteur de boot NT
```

On peut ensuite **rebooter** sous **NT**, sous réserve que les fichiers suivants figurent bien dans la racine :

- NTLDR
 - NTDETECT.COM
 - BOOT.INI
- et éventuellement NTBOOTDD.SYS si l'on boote sur un disque SCSI

NTLDR et *NTDETECT.COM* se trouvent dans le CDROM de NT (répertoire \I386)

BOOT.INI peut être édité comme indiqué plus haut

NTBOOTDD.SYS (éventuel) est une copie du driver livré avec la carte SCSI (p.ex. *AIC78XX.SYS*).

(La suite du lancement de NT est évidemment subordonnée à la présence des fichiers de NT dans la partition désignée dans le fichier `boot.ini` : existence du répertoire `\winnt`, de fichiers comme `ntoskrnl.exe` dans `\winnt\system32`, ...)



Comment installer DOS (6.22) après Windows 95 (Vers OSR1 uniquement)

Cette idée peut sembler à priori étrange, mais dans certains cas on peut avoir besoin de "revenir" sous **DOS** (ou Windows 3.x) pour tester ou modifier des applications fonctionnant sur d'autres plates-formes n'ayant pas Windows 95 installé.

NB : Cette opération peut être réalisée *sans outils particuliers, ni reformatage* de la partition. La seule condition est que la partition soit au format **FAT16** car DOS (6.22) ne sait reconnaître que FAT12 et FAT16, et en aucune façon FAT32.

Le mode opératoire est le suivant :

1. **En mode DOS sous 95, sauvegarder le secteur de boot (de 95 / 98)**
Pour cela, utiliser le fichier batch **saveboot.bat** (il est décrit plus haut) puis exécuter la commande suivante :

```
saveboot bootsect.w95
```

(le nom du fichier, p.ex. **bootsect.w95**, n'a aucune importance)

2. **Toujours en mode DOS 95, modifier (ou vérifier) MSDOS.SYS**
Dans le paragraphe [Options] doit figurer la ligne suivante :

```
BootMulti=1
```

3. **Rebooter avec une disquette DOS 6.22**
4. **Renommer les fichiers système de Windows 95**
*NB : La plupart de ces fichiers ont les attributs **R, H** et/ou **S***

Fichier **rensyst.bat**

```
@echo off
echo Renommage des fichiers systèmes de Windows 95
pause
c:
cd \
attrib -r -h -s *.*
ren MSDOS.SYS MSDOS.W40
ren IO.SYS WINBOOT.SYS
ren COMMAND.COM COMMAND.W40
ren AUTOEXEC.BAT AUTOEXEC.W40
ren CONFIG.SYS CONFIG.W40
```

5. **Copier les fichiers du DOS 6.22 dans un répertoire quelconque** (p.ex. c:\dos) puis exécuter la commande (depuis la disquette p.ex.)

```
sys c:
```

Elle place les fichiers systèmes **IO.SYS** et **MSDOS.SYS** de **DOS** en tête de la racine, mais elle **écrase** le secteur de **boot**, d'où l'utilité de l'avoir sauvegardé auparavant

6. **Restaurer le secteur de boot de Windows 95**
Pour cela, utiliser le fichier batch **restboot.bat** (il est décrit plus haut) puis exécuter la commande suivante :

```
restboot bootsect.w95
```

(reprendre le nom du fichier que l'on a donné avec **saveboot.bat**)

7. **Rebooter la machine**
Windows 95 redémarre, et le menu (actionné par F8) offre la possibilité de passer sous l'ancien OS (DOS 6.22)
Si l'on choisit Windows 95, on pourra alors constater que les fichiers du DOS (**IO.SYS**, **MSDOS.SYS**, **COMMAND.COM**,...) ont été renommés en **IO.DOS**, **MSDOS.DOS**, **COMMAND.DOS**,....

Les autres outils...

Si toutes ces manipulations ne vous inspirent que très modérément....

1. **Téléchargez** l'excellent "**BOOTPART**" (du à *Gilles VOLLANT*), qui va se charger de tout cela. C'est un exécutable 32 bits en mode console de 30 k, qui fonctionne aussi bien sous NT que sous DOS. Cet exécutable contient, entre autres, les 3 images de secteur de boot (DOS, 95, NT)
<http://www.winimage.com/bootpart.htm>
Ce logiciel a le statut (original!) de "cardware" (l'auteur demande simplement de lui envoyer une carte postale.)

```
> ....
> BOOTPART creates a 512 byte file that contains an image of a boot sector
> that loads the boot sector of the partition. That is, it creates a small
> program that loads and executes the specified partition's boot sector.
> ....
> Windows NT installs a boot sector that launches NTLDR and displays the
> Windows NT boot menu. If you lose this boot sector, Bootpart can restore
> it.
> Boot under MS-Dos (this can be MS-Dos 6.22 on your hard disk, on an MS-Dos
> bootable floppy, or the "MS-Dos 7.0" included in Windows 95 (you can obtain
> it by pressing Shift+F5 when Win95 loads). Then, enter the command :
> BOOTPART WINNT BOOT:C:
>
> If you want to remove the WinNT boot sector, you can replace it with the
> command "BOOTPART DOS622 BOOT:C:" or "BOOTPART WIN95 BOOT:C:"
> ...
```

Et plein d'autres possibilités !

2. **Faites l'acquisition** du très puissant outil "**System Commander**" ou "**Partition Magic**", spécialement **conçus** pour le *multi-boot*, qui autorisent n'importe quelle configuration, et dans n'importe quel ordre. Ceux sont deux produits commerciaux, au prix tout à fait raisonnable vu leurs possibilités (*environ 350 FF*).
(Pour plus d'informations, consultez le site de [V Communications ou Power Quest](#))
Ces deux outils créent leur propre **MBR**, et installe dans une partition (FAT) du disque un programme autonome qui est lancé lorsque le BIOS lit le MBR. Ce programme affiche sous forme de **menu** une liste des différents systèmes d'exploitation installés sur **les** disques (plus de de 100 systèmes sont reconnus : *DOS, Windows NT, Linux, SCO Unix, Windows 98, ...*). Avant même de lancer le système choisi, l'utilisateur peut, s'il le désire, éditer certains fichiers "texte" de boot, comme "boot.ini", "config.sys", ... System Commander ayant un éditeur pleine page intégré (indépendant de tout OS)
NB: la toute **dernière version** (dite "*Deluxe*") sait créer et **redimensionner** les partitions **sans perte** de données ! comme le fait **Partition Magic**.



Le fichier MSDOS.SYS (Windows 95 /98)

Windows 95 (98) comprend un fichier système nommé **MSDOS.SYS** placé dans la racine de la partition de boot.

Il possède les attributs *Lecture seule, Système et Caché (RHS)*.

Bien que possédant l'extension **SYS**, affectée habituellement à des fichiers binaires (système d'exploitation, drivers), c'est un **fichier texte**, possédant une structure analogue aux fichiers **.INI** de configuration de Windows (disparaissant peu à peu au profit de la Base de Registres)

Il est composé de 2 sections, contenant chacune les définitions de certains paramètres :

Section	Nom des paramètres	Signification / Valeurs possibles	Valeur par défaut
[Paths]	HostWinBootDrv	Disque de boot	C
	WinBootDir	Répertoire de Windows (emplacement des fichiers nécessaires à l'initialisation du système).	Le répertoire nécessaire au démarrage
	WinDir	Répertoire de Windows	Emplacement de Windows
[Options]	Autoscan (Spécifique Win 95 OSR2)	0 = Supprime le scandisk 1 = Valide le scandisk 2 = Lancement automatique si nécessaire et sans confirmation	0
	BootDelay	Temps pendant lequel le message Démarrage de Windows apparaît à l'écran.	2 (secondes)
	BootFailSafe	0 = Démarrage normal 1 = Démarrage en mode sans échec (1)	0
	BootGUI (GUI=Graphic User Interface)	0 = Désactive le chargement de l'interface graphique 1 = Chargement de l'interface Windows	1
	BootKeys	0 = Désactive les touches de fonctions au démarrage 1 = Utilisation des touches (F4, F5, F6, F8).	1
	BootMenu	0 = Désactive menu de démarrage (appui sur F8 pour l'avoir) 1 = Activation systématique du menu de démarrage.	0
	BootMenuDefault	Option par défaut du menu de démarrage.	1
	BootMenuDelay	Délai pour sélectionner une option du menu de démarrage	30 (secondes)
	BootMulti	0 = Désactive l'option de multi-boot 1 = Active les touches de fonctions F4 et F8	0
	BootWarn	0 = Désactive le message d'avertissement de mode sans échec 1 = Active le message	1
	BootWin	0 = Windows 95 n'est pas le système d'exploitation par défaut 1 = Force Windows 95 à se charger au démarrage.	1
	DoubleBuffer	0 = Double-buffering désactivé 1 = Double-buffering activé (pour les contrôleurs qui le nécessite, ex: contrôleurs SCSI).	0
	DBLSpace	0 = Non chargement de DBLSPACE.BIN. 1 = Chargement automatique de DBLSPACE.BIN.	1
	DRVSpace	0 = Non chargement de DRVSPACE.BIN. 1 = Chargement automatique de DRVSPACE.BIN.	1
	LoadTop	0 = Chargement COMMAND.COM / D??SPACE.BIN conventionnel 1 = Chargement COMMAND.COM / D??SPACE.BIN en haut des 640Ko.	1
Logo	0 = Logo animé de Windows95 désactivé 1 = Force le logo de Windows 95 à apparaître.	1	
Network	0 = Absence d'option réseau 1 = Réseau installé. Ajoute une option "Réseau" dans le menu de démarrage	0	

Raccourcis avec la touche "Windows" (claviers 105 touches)

<i>Combinaison de touche</i>	<i>Action</i>
<Win><Attn>	Affichage des propriétés système
<Win> E	Ouverture de l'explorateur
<Win> M	Réduction de toutes les fenêtres
<Maj><Win> M	Annulation réduction
<Win><F1>	Affichage Aide de Windows
<Win> F	Recherche de fichiers
<Ctrl><Win> F	Recherche sur le réseau
<Win><Tab>	Parcours des boutons barre des tâches

Configuration du panneau de configuration (Windows95/98)

Le lanceur (**control.exe**) du panneau de configuration de Windows 95/98 est une application **16 bits**, qui fait appel à un fichier **ini** (**control.ini**).

Il sert à passer le contrôle à différentes **DLL** (ayant l'extension **.cpl**) propres à un ou plusieurs matériels/fonctionnalités.

Il est possible de modifier certaines options du panneau de configuration à l'aide ce fichier, comme :

- **cacher (mais non pas supprimer) certains modules**
Control.ini contient une section intitulée **[don't load]** dans laquelle il suffit d'inclure des lignes ayant la syntaxe suivante :
<Nom_fichier.cpl>=NO

Exemple :

```
[don't load]
....
JOY.CPL=NO
....
```

NB : Les modules listés ci-dessous ne sont pas obligatoirement présents sur toutes les machine .

Fichier CPL	Rôle(s)
ACCESS.CPL	Accessibilité
APPWIZ.CPL	Ajout/suppression de programmes
DESK.CPL	Affichage
FINDFAST.CPL	Recherche accélérée
INET.CPL	Internet
INFRARED.CPL	Liaison infrarouge
INTL.CPL	Paramètres régionaux
JOY.CPL	Joystick
MAIN.CPL	Souris, Clavier, Polices, Imprimantes
MLCFG32.CPL	Courrier/Télécopies
MMSYS.CPL	Multimédia
MODEM.CPL	Modem
MSWEBCPL.CPL	Personal Web Server
NETCPL.CPL	Réseaux
ODBC32.CPL	ODBC (Open DataBase Connectivity)
PASSWORD.CPL	Mots de passe
POWERCFG.CPL	Alimentation
QW32.CPL	Quick Time
STICPL.CPL	Scanner et appareil photo-numérique
SYSDM.CPL	Système
TELEPHON.CPL	Numération téléphonique
THEMES.CPL	Thèmes du bureau
TIMEDATE.CPL	Date et heure
TWEAKUI.CPL	Personnalisation Windows (PowerToys)
WGPOCPL.CPL	Bureau de poste Microsoft Mail

Structure et paramètres du fichier "boot.ini" (Windows NT)

Le fichier **boot.ini** est lu au démarrage de NT par NTLDR. Chaque ligne de la section **[Operating system]** a l'une des structures suivantes :

<Nom ARC><Chemin>=<Libellé><Commutateurs>
<Racine DOS><Fichier secteur de boot>=<Libellé>[<Commutateurs>]

1ère structure

Elle concerne *Windows NT*

- **Nom ARC (Advanced RISC Computing)**

Un nom ARC, qui sert à désigner le disque et la partition où se trouve NT, est ainsi constitué :
SCSI(x)disk(y)rdisk(z)partition(w) (boot depuis un disque **SCSI**)

ou

MULTI(x)disk(y)rdisk(z)partition(w) (boot depuis un disque **IDE /EIDE /ESDI**)

La distinction SCSI ou MULTI est importante, car elle indique à NT comment procéder pour accéder aux premiers fichiers dont il a besoin :

- dans le cas de disque IDE, il va utiliser l'INT13h du BIOS,
- dans le cas de disque SCSI, il va utiliser un driver lié à la carte SCSI, ce driver s'appelant *NTBOOTDD.SYS* (copie p.ex. de *AIC78XX.SYS*, *AHA154X.SYS*, ...).

x	N° de contrôleur matériel SCSI dans l'ordre d'initialisation (BIOS) Certains disques SCSI peuvent apparaître aussi avec l'appellation MULTI en fonction du mode de gestion du BIOS
y	N° de bus pour les adaptateurs SCSI multibus (=ID SCSI) Toujours égal à 0 dans le cas de contrôleurs MULTI
z	N° de disque pour les contrôleurs MULTI (compris entre 0 et 3) Toujours égal à 0 pour les disques SCSI
w	N° de la partition (NB : commence à 1)

Exemples :

- Disque SCSI d' ID=3, avec 4 partitions, NT étant sur la 2ème, dans le répertoire *\wnt4*:
`scsi(0)disk(3)rdisk(0)partition(2)\WNT4="....."`

- Disque IDE "master" sur le 2ème connecteur IDE, 3 partitions, NT étant sur la 1ère, dans le répertoire *\winnt* :
`multi(0)disk(0)rdisk(2)partition(1)\WINNT="...."`

- **Chemin**
Le nom du répertoire, dans la partition considérée, dans lequel se trouve NT proprement dit.
- **Libellé**
Chaîne alphanumérique quelconque qui apparaîtra à l'écran dans le menu de choix d'OS
- **Commutateurs**
Facultatifs. Ils permettent de préciser le type d'exécution de NT.

<i>Commutateur</i>	<i>Signification</i>
/BASEVIDEO	Utilisation du driver standard d'affichage VGA. A utiliser dans le cas de changement de carte graphique
/BAUDRATE=nnnn	Spécifie la vitesse de transmission pour le debugging Par défaut, 9600 avec un modem et 19200 avec un null-modem Force le commutateur /DEBUG, même s'il n'a pas été précisé
/CRASHDEBUG	Charge le debugger , qui reste toutefois inactif tant qu'il n'y a pas d'erreur du noyau
/DEBUG	Charge le debugger , qui peut être activé à tout moment par une autre machine de debugging connectée à l'ordinateur. A utiliser en cas de problèmes répétitifs
/DEBUGPORT=comx	Spécifie le n° de port à utiliser pour le debugging . Force le commutateur /DEBUG, même s'il n'a pas été précisé
/MAXMEM:n	Spécifie le maximum de mémoire RAM que NT peut utiliser. A utiliser quand on suspecte une barrette RAM d'être défectueuse
/NODEBUG	Aucune information de debugging utilisée
/NOSERIALMICE=[COMx COMx,y,z,...]	Désactive la détection de souris sur le(s) port(s) série spécifié(s), ou sur tous les ports série si on ne précise aucun port.
/SOS	Affiche les noms de drivers au cours de chargement. A utiliser quand on pense qu'un driver est manquant ou défectueux

2ème structure

Elle concerne *DOS, Windows 95*

- **Racine DOS**
DOS et Windows 95 ne sachant pas démarrer depuis une unité autre que le premier disque dur ou la disquette, les seules valeurs possibles sont **C:** ou **A:**
- **Fichier_secteur_de_boot**
S'il n'y a pas d'ambiguïté, ce nom est **facultatif**. C'est le nom d'un fichier de 512 octets, qui est une image du secteur de boot de DOS ou de Windows 95. Ce nom est généralement **BOOTSECT.DOS**, mais ce n'est pas obligatoire. Tout autre nom peut convenir. Si DOS (ou Windows 95) est choisi par l'utilisateur, NTLDR lit ce fichier et le substitue (en mémoire, temporairement) au secteur de boot de NT, ce qui a pour conséquence de lancer le 1er fichier de l'OS correspondant (IO.SYS en principe)
- **Libellé**
Chaîne alphanumérique quelconque qui apparaîtra à **l'écran dans le menu de choix d'OS**
- **Commutateur**
S'il n'y a pas d'ambiguïté, ce commutateur est facultatif. On l'indiquera si l'on désire un **triple boot**, à savoir *NT, Windows 95* et *DOS*. Dans ce cas, la racine du disque C: contient 2 fichiers images de secteur de boot :
- **BOOTSECT.DOS** relatif à DOS (6.22 p.ex.)
- **BOOTSECT.W95** relatif à Windows 95
Les noms cités ici sont arbitraires. Suivant l'OS choisi (DOS ou Windows 95), NTLDR chargera le fichier image de secteur de boot correspondant. Le commutateur est à indiquer seulement si l'on souhaite que NT émule le processus de multi-boot de Windows 95 (actionnées en appuyant sur F8 lors du démarrage de 95). Dans ce cas, les valeurs qu'il peut prendre sont :
- **/win95dos** associé à la ligne de commande de DOS
- **/win95** associé à la ligne de commande de Windows 95



Exemple :Soit la configuration suivante :

- OS installés : **DOS 6.22, Windows 95, Windows NT 4, Windows NT 5** (bêta)

- 2 disques durs, ainsi partitionnés :

1. **2 partitions**

- la 1ère dédiée à DOS et Windows 95

- la 2ème dédiée à NT 4

2. **4 partitions**

- les 3 1ères dédiées à des applications et données

- la 4ème dédiée à NT 5

Le fichier **boot.ini** sera constitué comme suit :

(Temps d'attente de 20 secondes, choix de Windows 95 par défaut)

```
[boot loader]
timeout=20
default=C:\bootsect.w95
[Operating Systems]
c:\bootsect.622="L'ancetre (Dos 6.22)" /win95dos
c:\bootsect.w95="La glute a Billou" /win95
multi(0)disk(0)rdisk(0)partition(2)\WINNT="Un OS pour PC qui tient la route"
multi(0)disk(0)rdisk(0)partition(2)\WINNT="En cas de malheur sous NT..."
/basevideo /sos
multi(0)disk(0)rdisk(1)partition(4)\WINNT="Beta Windows NT Server 5.0"
multi(0)disk(0)rdisk(1)partition(4)\WINNT="Beta Windows NT Server 5.0 [VGA
mode]" /basevideo /sos
```

Création d'une disquette de boot partiel sous Windows NT

Tout d'abord, il faut préciser qu'il est **impossible** de créer une disquette bootable qui serait capable de contenir et charger tout le système d'exploitation de Windows NT, en raison de la taille même des fichiers exécutables et bibliothèques, ainsi que celle de la base de registres.

Par contre, il est possible de **commencer le démarrage depuis une disquette**, essentiellement dans le cas où l'un des **fichiers de départ** de NT est **défectueux** ou manquant, à savoir :

- **NTLDR** (le "loader" de NT)
- **BOOT.INI** (le fichier texte indiquant les systèmes d'exploitation disponibles)
- **NTDETECT.COM** (détermine le type de matériel installé)
- **BOOTSECT.DOS** (image du secteur de boot d'un autre OS, tel que DOS)
- **NTBOOTDD.SYS** (initialisation à partir d'un disque SCSI)

Pour créer une telle disquette, opérer ainsi :

1. **Formater** une disquette avec le gestionnaire de fichiers **NT** ou depuis la ligne de commande.
Ne pas utiliser de disquette préformatée DOS, car cette dernière a un secteur de boot prévu pour lancer IO.SYS et non pas NTLDR
2. **Copier** les fichiers (situés dans la racine de la partition de boot)
NTLDR
NTDETECT.COM
BOOT.INI
NTBOOTDD.SYS (seulement dans le cas de disque SCSI, et si le BIOS de la carte SCSI a été désactivé)
BOOTSECT.DOS (si l'on désire pouvoir redémarrer sous DOS)



Les séquences du démarrage de Windows NT

1. Le secteur de boot lance le programme **NTLDR**
2. **NTLDR** recherche les fichiers suivants :
 - BOOT.INI
 - NTDETECT.COM
 - NTBOODD.SYS (seulement dans le cas de disque SCSI, et si le BIOS de la carte SCSI a été désactivé)
 - BOOTSECT.DOS (éventuellement)
3. Il bascule le processeur en mode 386
4. Il lance un gestionnaire de fichiers très simple, basé sur l'INT13h (disque IDE) ou en utilisant NTBOODD.SYS (disque SCSI)
5. Il lit **BOOT.INI**, affiche les options correspondantes à l'écran et attend le choix de l'utilisateur
6. Si NT n'a pas été choisi, il charge le fichier BOOTSECT.DOS (ou un autre si le nom d'un fichier image de secteur de boot a été explicitement indiqué) à la place du secteur de boot initial, puis lui passe le contrôle
7. Si NT a été choisi, il lance **NTDETECT.COM** caractérisé par l'affichage à l'écran de "*NTDETECT Vxxx checking Hardware...*"
8. **NTDETECT.COM** inspecte :
 - le n° d'identification du PC
 - la carte vidéo
 - le type de clavier
 - les ports séries et parallèles
 - les lecteurs de disquettes
 - la souris (si elle existe)
9. Ensuite il crée la partie du **registre** concernant le **matériel**.
Ces données, **non permanentes**, peuvent se retrouver dans la section *HKEY_LOCAL_MACHINE\Hardware*
Cette section est donc reconstruite à chaque démarrage de l'ordinateur
10. Puis intervient le **lancement** du noyau :
 - Chargement du "**HAL**" (Hardware Abstract Layer), qui permet au système d'être indépendant du matériel, et de **NTOSKRNL**, qui va lire les données situées dans *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services*, afin de déterminer les drivers et services à charger possédant un statut de démarrage "**amorcé**" (cf Panneau de configuration/Périphériques).
Cette phase est caractérisée par l'affichage à l'écran de "*OS Loader Vxxx*", chaque point correspondant à un pilote.
11. **Initialisation** du noyau
L'écran devient bleu et passe en mode 50 lignes, avec affichage d'un message comme "*Microsoft Windows NT Version 4...*"
Le noyau inspecte à nouveau la clef *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services*, pour les pilotes possédant un statut de démarrage "**système**". Cette phase est caractérisée par l'affichage à l'écran d'une suite de points, chaque point correspondant à un pilote. Un nouveau "*CurrentControlSet*" est construit, mais non sauvegardé.
12. Chargement des **services**
Le gestionnaire de services (SMSS.EXE) est lancé, charge le sous-système Win32, et les services possédant un statut de démarrage "**automatique**". Un nouveau "*CurrentControlSet*" est construit
13. Lancement du **sous-système Windows**
Une copie de "*CurrentControlSet*" est copiée dans "Dernière bonne configuration connue", WINLOGON.EXE est lancé, lequel inspecte la clef *HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Winlogon* et recherche la valeur de l'entrée *System*, qui contient les noms des sous-systèmes (p.ex. ISASS.EXE, gestionnaire de sécurité locale)
14. A ce moment apparaît (enfin!) la boîte de dialogue invitant à appuyer sur CTRL-ALT-SUPP pour **démarrer une session**

