

V3. Exemple de fichier de sauvegarde

Considérons ces deux fichiers :

f1.txt contenant le texte :

```
une ligne
```

f2.txt contenant le texte :

```
Une première ligne  
Une seconde  
Une troisième  
Une quatrième  
Et enfin la dernière
```

Une fois ces fichiers ajoutés au SF, l'affichage du SF généré donnera ceci :

```
Sf de nom sda, super bloc :  
taille bloc = 64, date der modif = Sat Nov  8 18:04:22 2025  
Inodes :  
  
-----Inode----[0] :  
type : Ordinaire  
  taille : 10 octets  
  date dernier accès : Sat Nov  8 18:04:22 2025  
  date dernière modification : Sat Nov  8 18:04:22 2025  
  date dernière modification inode : Sat Nov  8 18:04:22 2025  
  Données :  
une ligne  
  
-----Inode----[1] :  
type : Ordinaire  
  taille : 84 octets  
  date dernier accès : Sat Nov  8 18:04:22 2025  
  date dernière modification : Sat Nov  8 18:04:22 2025  
  date dernière modification inode : Sat Nov  8 18:04:22 2025  
  Données :  
Une première ligne  
Une seconde  
Une troisième  
Une quatrième  
Et enfin la dernière
```

Supposons que l'on utilise la fonction SauvegarderSF() pour sauvegarder ce SF sur disque, dans un fichier de nom « save_v3 ».

On pourra ensuite visualiser le contenu de ce fichier grâce à différentes commandes Unix.

```
cat save_v3
```

```
sdaaaaaaaaaaaaaaaaaaaaaaasi
sisisiune ligne
TsisisiUne première ligne
Une seconde
Une troisième
Une quatrième
Et enfin la dernière
```

Comme le fichier de sauvegarde est un fichier « binaire », certains caractères ne seront pas « lisibles ». Ils correspondent ici aux données sauvegardées (dates, tailles, etc.) autres que le contenu (texte) des blocs.

On peut visualiser ce même fichier de façon différente grâce à la commande `od` (voir son man pour plus de détails).

```
od -t d1 save_v3
```

```
0000000 115 100 97 0 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
0000020 -66 -66 -66 -66 -66 -66 -66 -66 -66 6 115 15 105 0 0 0
0000040 0 2 0 0 0 0 0 0 0 0 0 0 0 10 0 0
0000060 0 0 0 0 0 6 115 15 105 0 0 0 0 6 115 15
0000100 105 0 0 0 0 6 115 15 105 0 0 0 0 1 0 0
0000120 0 117 110 101 32 108 105 103 110 101 10 1 0 0 0 0
0000140 0 0 0 84 0 0 0 0 0 0 0 6 115 15 105 0
0000160 0 0 0 6 115 15 105 0 0 0 0 6 115 15 105 0
0000200 0 0 0 2 0 0 0 85 110 101 32 112 114 101 109 105
0000220 -61 -88 114 101 32 108 105 103 110 101 10 85 110 101 32 115
0000240 101 99 111 110 100 101 10 85 110 101 32 116 114 111 105 115
0000260 105 -61 -88 109 101 10 85 110 101 32 113 117 97 116 114 105
0000300 -61 -88 109 101 10 69 116 32 101 110 102 105 110 32 108 97
0000320 32 100 101 114 110 105 -61 -88 114 101 10
0000333
```

Et on voit que les caractères « lisibles » apparaissent sous leur code de la table ASCII exprimé en hexadécimal. Le premier caractère possède le code hexadécimal 115, qui correspond au code de « s ».

Enfin, la commande `hexedit` permet de visualiser le contenu sous les deux formes (voir man pour plus de détails). On peut sortir de cette commande avec `Ctrl-C`.

```
hexedit save_v3
```

```
00000000 |73 64 61 00 BE BE BE BE BE BE BE BE BE BE BE BE BE BE BE BE BE BE BE sda.....
00000018 BE 06 73 0F 69 00 00 00 00 02 00 00 00 00 00 00 00 00 0A 00 00 ..s.i.....
00000030 00 00 00 00 00 06 73 0F 69 00 00 00 00 06 73 0F 69 00 00 00 .....s.i.....s.
00000048 69 00 00 00 00 01 00 00 00 75 6E 65 20 6C 69 67 6E 65 0A 01 00 00 00 00 i.....une ligne....
00000060 00 00 00 54 00 00 00 00 00 00 00 06 73 0F 69 00 00 00 00 06 73 0F 69 00 ...T.....s.i.....s.i.
00000078 00 00 00 06 73 0F 69 00 00 00 00 02 00 00 00 55 6E 65 20 70 72 65 6D 69 ...s.i.....Une premi
00000090 C3 A8 72 65 20 6C 69 67 6E 65 0A 55 6E 65 20 73 65 63 6F 6E 64 65 0A 55 ..re ligne.Une seconde.U
000000A8 6E 65 20 74 72 6F 69 73 69 C3 A8 6D 65 0A 55 6E 65 20 71 75 61 74 72 69 ne troisi..me.Une quatri
000000C0 C3 A8 6D 65 0A 45 74 20 65 6E 66 69 6E 20 6C 61 20 64 65 72 6E 69 C3 A8 ..me.Et enfin la derni..
000000D8 72 65 0A
000000F0
```

Les codes ASCII des caractères « lisibles » apparaissent ici en partie gauche et en décimal. 73 est le code ASCII décimal du caractère « s ». Dans la partie droite, on a les caractères lisibles qui sont visualisés et les autres qui apparaissent comme des « . » (ou parfois comme des caractères lisibles qui n'en sont pas).

Tout cela pour vous montrer que votre fichier de sauvegarde ne sera pas entièrement lisible pour vous (comme un fichier exécutable qui est lui aussi un fichier binaire) et que seule la restauration permettra de savoir s'il est bien formé.