

OricFS

Table des matières

Présentation.....	2
Installation.....	3
Utilisation.....	3
Affichage du contenu d'un fichier :.....	5
Conversion automatique.....	5
Conversion forcée.....	6
Dump d'un fichier : ?dump.....	6
Code source d'un programme assembleur : ?asm.....	6
Code source d'un programme BASIC : ?basic.....	7
Comment obtenir des informations concernant un fichier ?.....	8
Informations concernant le fichier : ?info.....	8
Fichiers spéciaux:	9
Obtenir des informations concernant l'image courante.....	9
Changer l'image disque courante.....	11
Créer une nouvelle image vierge.....	11
Accès direct à un secteur de la disquette.....	11
Fonctions spéciales:	13
Extraction d'un programme BASIC sous forme de fichier source.....	13
Transfert d'un programme source BASIC.....	13
Création d'un fichier '.tap'.....	13
Transfert du contenu d'un fichier '.tap'.....	13
Affichage du contenu d'un fichier de données	14
Fichiers ARY.....	15
Transfert d'un programme BASIC d'une image SEDORIC vers une image FT-DOS.	15
Format de fichier supportés.....	16
FTDOS.....	16
SEDORIC.....	16
Utilitaires.....	17
Txt2bas / bas2txt.....	17
bin2tap.....	18
Annexes.....	19
Exemples de fichiers SCR.....	19
Exemples de fichiers HLP.....	20
Exemple de fichier WIN.....	22

Index des illustrations

Illustration 1: WS.SCR.....	6
Illustration 2: CAT.SCR.....	18
Illustration 3: LOAD.SCR.....	18
Illustration 4: BDDISK.HLP.....	19
Illustration 5: V30NEWS02.HLP.....	19
Illustration 6: V30NEWS05.HLP.....	20
Illustration 7: ADDRESS.WIN.....	21

Présentation

OricFS est un utilitaire écrit en Python permettant de manipuler le contenu d'une image '**.ask**' en la « montant » comme n'importe quel filesystem reconnu par Linux.

Il s'appuie sur le module FUSE du noyau.

Il supporte actuellement les images FT-DOS 3.2 et SEDORIC en lecture et écriture.

Il est composé d'un programme principal et de plugins :

- **Programme principal**
 - OricFS-0.3.1-py2.6.egg
- **DOS**
 - OricFS_FTDOS-0.2-py2.6.egg : Prise en charge des images FT-DOS
 - OricFS_Sedoric-0.4-py2.6.egg : Prise en charge des images SEDORIC
- **Langages**
 - OricFS_ASM-0.1-py2.6.egg : Conversion fichier binaire → source assembleur
 - OricFS_BASIC-0.3.1-py2.6.egg : Conversion fichier BASIC → source texte
- **Utilitaires**
 - OricFS_Info-0.1-py2.6.egg : Permet d'obtenir des informations sur un fichier
 - OricFS_Tape-0.2-py2.6.egg : Conversion depuis et vers un fichier « .tap »

Il existe deux versions, une pour Python 2.6 et l'autre pour Python 2.7. La différence vient de la version de la librairie fuse-python fournie avec ces versions (au moins pour Ubuntu 14.04) mais il est possible que la version pour Python 2.6 puisse fonctionner avec un Python 2.7.

Les packages ont été générés avec un OS 64 bits.

Installation

Le programme nécessite **Python** version 2.6 ou supérieure (mais non testé avec une version 3.x) et le module **fuse-python** version 0.2 ainsi que les packages **fuse** et **setuptools** de votre distribution.

Récupérer le fichier **OricFS-0.3.1-py2.6.egg** et le copier dans le répertoire courant.

Puis taper la commande :

```
easy_install ./OricFS-0.3.1-py2.6.egg
```

Il faut également installer au moins un module DOS :

```
easy_install ./OricFS_FTDOS-0.2-py2.6.egg
```

Les autres plugins sont facultatifs.

Utilisation

Usage:

```
OricFS: a filesystem to allow mounting a Oric disk image
oricfs.py [mountpoint] [options]
```

Options:

```
--version          show program's version number and exit
-h, --help         show this help message and exit
-o opt,[opt...]    mount options
-o source=PATH     source path to mount [default: none]
-i ICON, --icon=ICON  Icon [default: drive-harddisk-usb]
-v, --verbose      Verbose mode [default: False]
```

Exemples :

- Montage de l'image **/Oric/Disks/dos/FTDOS.dsk** sur le point de montage **./fs**

```
mkdir ./fs
oricfs.py /Oric/Disks/dos/FTDOS.dsk ./fs
```
- Lancement sans préciser d'image (voir page 11 pour plus d'information)

```
mkdir ./fs
oricfs.py ./fs
```

Il est également possible de lancer le programme en mode debug

```
oricfs.py -d /Oric/Disks/dos/FTDOS.dsk ./fs 2>err|less
```

A partir de maintenant, le répertoire **./fs** peut être utilisé pour manipuler le contenu de l'image disque.

```
$ cd fs
```

```
$ ls -l
```

```
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 AUTO .SCR
-rw-rw-rw- 1 hcl users 512 janv. 26 19:35 AUTODOS .BAS
-rw-rw-rw- 1 hcl users 2304 janv. 26 19:35 BKP .BAS
-rw-rw-rw- 1 hcl users 512 janv. 26 19:35 BKP .CMD
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 BKP .SCR
```

```

-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 CAT .SCR
-rw-rw-rw- 1 hcl users 512 janv. 26 19:35 COPY .CMD
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 COPY .SCR
-rw-rw-rw- 1 hcl users 1024 janv. 26 19:35 COPY1 .BAS
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 COPY1 .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 CUT .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 DEL .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 DEMOUNT .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 DNAME .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 DS .SCR
-rw-rw-rw- 1 hcl users 2560 janv. 26 19:35 FORMAT .BAS
-rw-rw-rw- 1 hcl users 768 janv. 26 19:35 FORMAT .CMD
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 FORMAT .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 FS .SCR
-rw-rw-rw- 1 hcl users 15872 janv. 26 19:35 FTDOS3-2.SYS
-rw-rw-rw- 1 hcl users 512 janv. 26 19:35 INIT .BAS
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 INIT .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 LOAD .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 LOCK .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 MASTER .SCR
-rw-rw-rw- 1 hcl users 1024 janv. 26 19:35 MATEST .BAS
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 MERGE .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 MOUNT .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 RENAME .SCR
-rw-rw-rw- 1 hcl users 6400 janv. 26 19:35 REPERTOR.BAS
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 RS .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 SAVE .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 SEARCH .SCR
-rw-rw-rw- 1 hcl users 768 janv. 26 19:35 SECTOR .DAT
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 START .SCR
-rw-rw-rw- 1 hcl users 512 janv. 26 19:35 TEST .BAS
-rw-rw-rw- 1 hcl users 1792 janv. 26 19:35 TESTDIR .BAS
-rw-rw-rw- 1 hcl users 768 janv. 26 19:35 TESTSEQU.BAS
-rw-rw-rw- 1 hcl users 512 janv. 26 19:35 TKD .BAS
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 TKD .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 TKDA .BAS
-rw-rw-rw- 1 hcl users 1792 janv. 26 19:35 TKDA .CMD
-rw-rw-rw- 1 hcl users 1280 janv. 26 19:35 TKDO .BAS
-rw-rw-rw- 1 hcl users 1792 janv. 26 19:35 TKDO .CMD
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 UNLOCK .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 UNSTART .SCR
-rw-rw-rw- 1 hcl users 1536 janv. 26 19:35 WS .SCR

```

Par défaut, les fichiers sont traités comme des fichiers binaires pour toutes les commandes Linux. La commande '**chmod**' est supportée pour le (dé)verrouillage en lecture seule s'un fichier

Le « démontage » se fait par la commande **fusermount** comme cela est indiqué au lancement du programme :

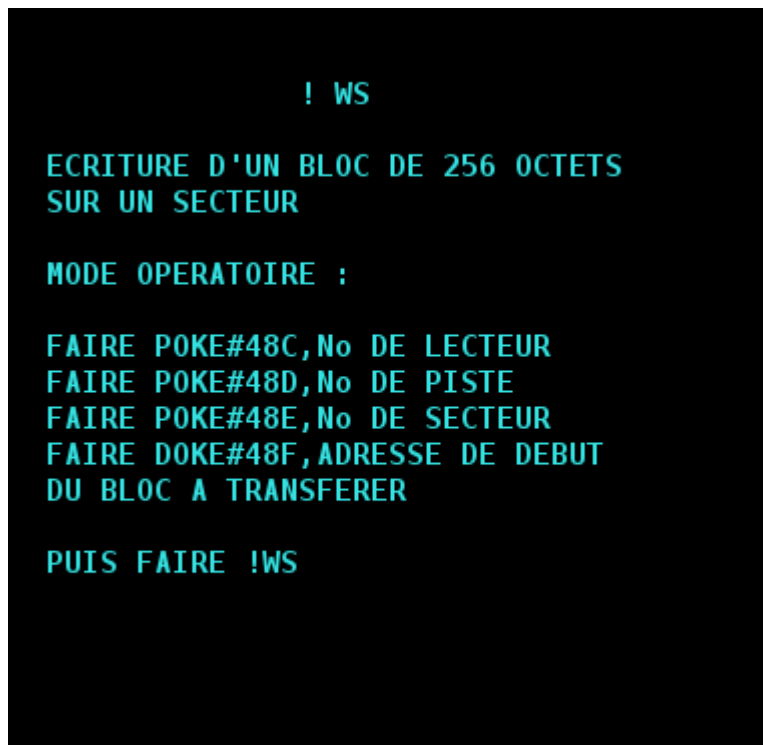
```
$ fusermount -u ./fs
```

Affichage du contenu d'un fichier :

Conversion automatique

Si on souhaite afficher le contenu d'un fichier dans un format compréhensible, il faut ajouter un '?' à la fin du nom du fichier:

```
$ cat WS\ \ \ \ \ \ .SCR?
```



```
! WS

ECRITURE D'UN BLOC DE 256 OCTETS
SUR UN SECTEUR

MODE OPERATOIRE :

FAIRE POKE#48C, No DE LECTEUR
FAIRE POKE#48D, No DE PISTE
FAIRE POKE#48E, No DE SECTEUR
FAIRE DOKE#48F, ADRESSE DE DEBUT
DU BLOC A TRANSFERER

PUIS FAIRE !WS
```

Illustration 1: WS.SCR

```
$ cat AUTODOS\ .BAS?
 1 !MASTER"FTDOS3-2"
 5 POKE#48C,1
10 FOR J=4 TO 6
15 POKE#48D,J
20 FOR I=1 TO 17
30 POKE#48E,I
40 !DS
50 NEXT I
60 NEXT J
70 POKE#48D,J
80 FOR I=1 TO 14
90 POKE#48E,I
100 !DS
110 NEXT I
120 PRINT"FTDOS MIS EN PLACE"
```

Conversion forcée

La plupart des formats des fichiers sont reconnus automatiquement mais il est possible de préciser un format autre que celui par défaut le précisant après le '?'

Si le nom du format n'est pas reconnu, il sera remplacé par un 'dump'

Dump d'un fichier : ?dump

```
$ cat AUTODOS\ .BAS?dump
0500  00 17 05 01 00 C0 4D 41 53 54 45 52 22 46 54 44      .....MASTER"FTD
0510  4F 53 33 2D 32 22 00 23 05 05 00 B9 23 34 38 43      OS3-2".#....#48C
0520  2C 31 00 31 05 0A 00 8D 20 4A D4 34 20 C3 20 36      ,1.1.... J.4 . 6
0530  00 3D 05 0F 00 B9 23 34 38 44 2C 4A 00 4C 05 14      .=....#48D,J.L..
0540  00 8D 20 49 D4 31 20 C3 20 31 37 00 58 05 1E 00      .. I.1 . 17.X...
0550  B9 23 34 38 45 2C 49 00 60 05 28 00 C0 44 53 00      .#48E,I.`.(..DS.
0560  68 05 32 00 90 20 49 00 70 05 3C 00 90 20 4A 00      h.2.. I.p.<.. J.
0570  7C 05 46 00 B9 23 34 38 44 2C 4A 00 8B 05 50 00      |.F..#48D,J...P.
0580  8D 20 49 D4 31 20 C3 20 31 34 00 97 05 5A 00 B9      . I.1 . 14...Z..
0590  23 34 38 45 2C 49 00 9F 05 64 00 C0 44 53 00 A7      #48E,I...d..DS..
05A0  05 6E 00 90 20 49 00 C1 05 78 00 BA 22 46 54 44      .n.. I...x..."FTD
05B0  4F 53 20 4D 49 53 20 45 4E 20 50 4C 41 43 45 22      OS MIS EN PLACE"
05C0  00 00 00      ...
```

Code source d'un programme assembleur : ?asm

Peut poser des problèmes si le programme contient aussi des données.

```
$ cat BKP\ \ \ \ \ .CMD?asm
// File: BKP      .CMD
// Size: 115
// .word $b300
* = $b300

      B300  A9 3C      LDA #$3C
      B302  8D FE FF  STA 6502_IRQVector
      B305  A9 B3      LDA #$B3
      B307  8D FF FF  STA 6502_IRQVector_H
      B30A  A9 98      LDA #$98
      B30C  8D F4 03  STA jsm_Command
      B30F  A0 00      LDY #$00
      B311  A2 11      LDX #$11
      B313  58          CLI
      B314  C0 00      CPY #$00
      B316  F0 FC      BEQ $B314
      B318  C0 00      CPY #$00
      B31A  D0 FC      BNE $B318
      B31C  A9 1C      LDA #$1C
      B31E  2D F4 03  AND jsm_Status/Command
      B321  D0 13      BNE $B336
      B323  E6 01      INC $01
      B325  CA          DEX
      B326  D0 EC      BNE $B314
      B328  A9 0E      LDA #$0E
      B32A  2D F4 03  AND jsm_Status/Command
      B32D  D0 07      BNE $B336
      B32F  A9 D0      LDA #$D0
      B331  8D F4 03  STA jsm_Command
```

```

B334 A9 00 LDA #$00
B336 8D FF B3 STA $B3FF
B339 4C 22 04 JMP FTDOS_Activate_BASIC

B33C AD F7 03 LDA jsm_Data
B33F 91 00 STA ($00),Y
B341 C8 INY
B342 40 RTI

B343 B1 00 LDA ($00),Y
B345 8D F7 03 STA jsm_Data
B348 C8 INY
B349 40 RTI

B34A A9 00 LDA #$00
B34C 8D 20 04 STA $0420
B34F A9 B3 LDA #$B3
B351 8D 21 04 STA $0421
B354 4C 14 04 JMP FTDOS_Activate_OVERLAY_&_Execute_JMP($420)

B357 A9 64 LDA #$64
B359 8D 20 04 STA $0420
B35C A9 B3 LDA #$B3
B35E 8D 21 04 STA $0421
B361 4C 14 04 JMP FTDOS_Activate_OVERLAY_&_Execute_JMP($420)

B364 A9 43 LDA #$43
B366 8D FE FF STA 6502_IRQVector
B369 A9 B3 LDA #$B3
B36B 8D FF FF STA 6502_IRQVector_H
B36E A9 BA LDA #$BA
B370 4C 0C B3 JMP $B30C

```

Remarque :

Préciser une conversion pour ce type de fichier est en fait inutile, `cat BKP\ \ \ \ \ .CMD?` a le même effet.

Code source d'un programme BASIC : ?basic

```

$ cat AUTODOS\ .BAS?basic
 1 !MASTER"FTDOS3-2"
 5 POKE#48C,1
10 FOR J=4 TO 6
15 POKE#48D,J
20 FOR I=1 TO 17
30 POKE#48E,I
40 !DS
50 NEXT I
60 NEXT J
70 POKE#48D,J
80 FOR I=1 TO 14
90 POKE#48E,I
100 !DS
110 NEXT I
120 PRINT"FTDOS MIS EN PLACE"

```

Remarque :

Préciser une conversion pour ce type de fichier est en fait inutile, `cat AUTODOS\ .BAS?` a le même effet.

Comment obtenir des informations concernant un fichier ?

Informations concernant le fichier : ?info

```
$ cat AUTODOS\ .BAS?info
;
; Informations for: AUTODOS .BAS
;
Sector          : 16
Track           : 23
Lock            : U
Start           : 0x0500 ( 1280 )
Size            : 0x00C3 ( 195 )
Content_type    : basic
Type            : S
Side            : 0
Size            : 2
```

Fichiers spéciaux:

Obtenir des informations concernant l'image courante

Le fichier `.source` contient des informations sur le type et le contenu de la disquette.

```
$ cat .source
;
; Disk image informations
;
[Default]
Source      : /Oric/Disks/dos/FTDOS.dsk
Volume     : MASTER
DOS        : FTDOS
Files      : 51
Size       : 1394
Free       : 960
Sectors    : 17
Tracks     : 41
Sides      : 2
Sectorsize : 256
Geometry   : 1
Offset     : 256
Dirs       : {'FORTH .BIN': {'sector': 8, 'track': 29, 'lock': 'U',
'content_type': 'asm', 'type': 'S', 'side': 0, 'size': 33}, 'LOAD .SCR':
{'sector': 14, 'track': 15, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'UNLOCK .SCR': {'sector': 7, 'track': 18, 'lock': 'U',
'content_type': 'lscreen', 'type': 'S', 'side': 0, 'size': 6}, 'FORTH-FD.COMD':
{'sector': 9, 'track': 27, 'lock': 'U', 'content_type': 'asm', 'type': 'S',
'side': 0, 'size': 38}, 'TKD .BAS': {'sector': 3, 'track': 11, 'lock': 'U',
'content_type': 'basic', 'type': 'S', 'side': 0, 'size': 2}, 'FORTH0 .BIN':
{'sector': 7, 'track': 31, 'lock': 'U', 'content_type': 'asm', 'type': 'S',
'side': 0, 'size': 2}, 'MOUNT .SCR': {'sector': 11, 'track': 20, 'lock': 'U',
'content_type': 'lscreen', 'type': 'S', 'side': 0, 'size': 6}, 'COPY1 .BAS':
{'sector': 13, 'track': 12, 'lock': 'U', 'content_type': 'basic', 'type': 'S',
'side': 0, 'size': 4}, 'TESTDIR .BAS': {'sector': 4, 'track': 13, 'lock': 'U',
'content_type': 'basic', 'type': 'S', 'side': 0, 'size': 7}, 'AUTO .SCR':
{'sector': 1, 'track': 14, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'FORTH-OR.BIN': {'sector': 11, 'track': 25, 'lock': 'U',
'content_type': 'asm', 'type': 'S', 'side': 0, 'size': 32}, 'MERGE .SCR':
{'sector': 7, 'track': 14, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'INIT .BAS': {'sector': 9, 'track': 25, 'lock': 'U',
'content_type': 'basic', 'type': 'S', 'side': 0, 'size': 2}, 'BKP .BAS':
{'sector': 9, 'track': 10, 'lock': 'U', 'content_type': 'basic', 'type': 'S',
'side': 0, 'size': 9}, 'FORMAT .SCR': {'sector': 2, 'track': 19, 'lock': 'U',
'content_type': 'lscreen', 'type': 'S', 'side': 0, 'size': 6}, 'RS .SCR':
{'sector': 7, 'track': 22, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'TEST .BAS': {'sector': 6, 'track': 67, 'lock': 'U',
'content_type': 'basic', 'type': 'S', 'side': 0, 'size': 2}, 'COPY .CMD':
{'sector': 17, 'track': 12, 'lock': 'U', 'content_type': 'asm', 'type': 'S',
'side': 0, 'size': 2}, 'UNSTART .SCR': {'sector': 1, 'track': 22, 'lock': 'U',
'content_type': 'lscreen', 'type': 'S', 'side': 0, 'size': 6}, 'BKP .SCR':
{'sector': 6, 'track': 21, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'AUTODOS .BAS': {'sector': 16, 'track': 23, 'lock': 'U',
'content_type': 'basic', 'type': 'S', 'side': 0, 'size': 2}, 'COPY1 .SCR':
```

```
{'sector': 10, 'track': 17, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'SECTOR .DAT': {'sector': 6, 'track': 10, 'lock': 'U',
'content_type': 'data', 'type': 'S', 'side': 0, 'size': 3}, 'CAT .SCR':
{'sector': 9, 'track': 16, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'TKDO .BAS': {'sector': 11, 'track': 11, 'lock': 'U',
'content_type': 'basic', 'type': 'S', 'side': 0, 'size': 5}, 'FORMAT .CMD':
{'sector': 3, 'track': 10, 'lock': 'U', 'content_type': 'asm', 'type': 'S',
'side': 0, 'size': 3}, 'LOCK .SCR': {'sector': 1, 'track': 18, 'lock': 'U',
'content_type': 'lscreen', 'type': 'S', 'side': 0, 'size': 6}, 'FTDOS3-2.SYS':
{'sector': 0, 'track': 0, 'lock': 'U', 'content_type': 'asm', 'type': 'S',
'side': 0, 'size': 62}, 'SEARCH .SCR': {'sector': 2, 'track': 15, 'lock': 'U',
'content_type': 'lscreen', 'type': 'S', 'side': 0, 'size': 6}, 'DEMOUNT .SCR':
{'sector': 17, 'track': 20, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'INIT .SCR': {'sector': 8, 'track': 19, 'lock': 'U',
'content_type': 'lscreen', 'type': 'S', 'side': 0, 'size': 6}, 'START .SCR':
{'sector': 12, 'track': 21, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'TKDA .BAS': {'sector': 5, 'track': 11, 'lock': 'U',
'content_type': 'basic', 'type': 'S', 'side': 0, 'size': 6}, 'MATEST .BAS':
{'sector': 14, 'track': 13, 'lock': 'U', 'content_type': 'basic', 'type': 'S',
'side': 0, 'size': 4}, 'DEL .SCR': {'sector': 3, 'track': 16, 'lock': 'U',
'content_type': 'lscreen', 'type': 'S', 'side': 0, 'size': 6}, 'MASTER .SCR':
{'sector': 14, 'track': 19, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'SAVE .SCR': {'sector': 8, 'track': 15, 'lock': 'U',
'content_type': 'lscreen', 'type': 'S', 'side': 0, 'size': 6}, 'WS .SCR':
{'sector': 10, 'track': 23, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'FORMAT .BAS': {'sector': 1, 'track': 4, 'lock': 'U',
'content_type': 'basic', 'type': 'S', 'side': 0, 'size': 10}, 'BKP .CMD':
{'sector': 1, 'track': 11, 'lock': 'U', 'content_type': 'asm', 'type': 'S',
'side': 0, 'size': 2}, 'REPEDITOR.BAS': {'sector': 1, 'track': 24, 'lock': 'U',
'content_type': 'basic', 'type': 'S', 'side': 0, 'size': 25}, 'TKDO .CMD':
{'sector': 6, 'track': 12, 'lock': 'U', 'content_type': 'asm', 'type': 'S',
'side': 0, 'size': 7}, 'TESTSEQU.BAS': {'sector': 11, 'track': 13, 'lock': 'U',
'content_type': 'basic', 'type': 'S', 'side': 0, 'size': 3}, 'DNAME .SCR':
{'sector': 5, 'track': 20, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'RENAME .SCR': {'sector': 15, 'track': 16, 'lock': 'U',
'content_type': 'lscreen', 'type': 'S', 'side': 0, 'size': 6}, 'DS .SCR':
{'sector': 13, 'track': 22, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'CUT .SCR': {'sector': 13, 'track': 14, 'lock': 'U',
'content_type': 'lscreen', 'type': 'S', 'side': 0, 'size': 6}, 'COPY .SCR':
{'sector': 4, 'track': 17, 'lock': 'U', 'content_type': 'lscreen', 'type': 'S',
'side': 0, 'size': 6}, 'FS .SCR': {'sector': 4, 'track': 23, 'lock': 'U',
'content_type': 'lscreen', 'type': 'S', 'side': 0, 'size': 6}, 'TKDA .CMD':
{'sector': 16, 'track': 11, 'lock': 'U', 'content_type': 'asm', 'type': 'S',
'side': 0, 'size': 7}, 'TKD .SCR': {'sector': 13, 'track': 18, 'lock': 'U',
'content_type': 'lscreen', 'type': 'S', 'side': 0, 'size': 6}}
```

Les valeurs des champs 'size' et 'Free' sont exprimées en nombre de secteurs.

Le champs 'dirs' contient la liste de tous les fichiers présents sur la disquette ainsi que des informations sur leurs taille, attributs, emplacement,...

Exemple :

```
'LOAD .SCR': {'sector': 14, 'track': 15, 'lock': 'U', 'content_type':
'lscreen', 'type': 'S', 'side': 0, 'size': 6}
```

Se lit :

'LOAD .SCR'	Nom du fichier
'sector': 14, 'track': 15	Emplacement du FCB
'lock': 'U'	Fichier non verrouillé
'content_type': 'lscreen'	Fichier de type 'écran texte'
'type': 'S'	Fichier de type séquentiel
'side': 0	FCB sur la face 0

'size': 6

Taille du fichier en secteurs

Changer l'image disque courante

```
$ echo "/path/to/my/img.dsk" > .source
```

Créer une nouvelle image vierge

Il faut écrire les mêmes informations que celles renvoyées par `cat .source`

Remarques:

- Ceci ne fonctionne pour le moment que pour une disquette FTDOS
- Le paramètre 'sectorsize' n'est pas pris en compte
- La nouvelle image devient l'image courante
- Si le fichier de destination, ici `/tmp/test.dsk`, existe déjà, il est écrasé sans ménagement !!!

Exemple:

pour créer une nouvelle image vierge '`/tmp/test.dsk`' de type FTDOS et ayant pour nom "TEST-01"

- Editer un fichier "`/tmp/disk.info`" contenant:

```
[Default]
Source      : /tmp/test.dsk
Volume     : TEST-01
DOS        : FTDOS
Sectors    : 17
Tracks     : 41
Sides      : 2
Sectorsize : 256
```

- Ensuite, un simple:

```
$ cat /tmp/disk.info > .source
```

Et voilà!

Accès direct à un secteur de la disquette

`.t<xx>s<yy>`

xx: Numéro de piste

yy: Numéro de secteur

`.h<xx>t<yy>s<zz>`

xx: Numéro de la face (0 ou 1)

yy: Numéro de piste

zz: Numéro de secteur

Seule la lecture est autorisée pour le moment.

Exemple:

voir le contenu du premier secteur du catalogue d'une disquette FTDOS

```
$ cat .t20s2?
0000 00 00 0D 02 00 00 55 46 54 44 4F 53 33 2D 32 2E .....UFTDOS3-2.
0010 53 59 53 53 3E 00 04 01 55 46 4F 52 4D 41 54 20 SYSS>...UFORMAT
0020 20 2E 42 41 53 53 0A 00 43 06 55 54 45 53 54 20 .BASS..C.UTEST
0030 20 20 20 2E 42 41 53 53 02 00 0A 03 55 46 4F 52 .BASS....UFOR
0040 4D 41 54 20 20 2E 43 4D 44 53 03 00 0A 06 55 53 MAT .CMDS....US
0050 45 43 54 4F 52 20 20 2E 44 41 54 53 03 00 0A 09 ECTOR .DATS....
0060 55 42 4B 50 20 20 20 20 20 2E 42 41 53 53 09 00 UBKP .BASS..
0070 0B 01 55 42 4B 50 20 20 20 20 20 2E 43 4D 44 53 ..UBKP .CMDS
0080 02 00 0B 03 55 54 4B 44 20 20 20 20 20 2E 42 41 ....UTKD .BA
0090 53 53 02 00 0B 05 55 54 4B 44 41 20 20 20 20 2E SS....UTKDA .
00A0 42 41 53 53 06 00 0B 0B 55 54 4B 44 4F 20 20 20 BASS....UTKDO
00B0 20 2E 42 41 53 53 05 00 0B 10 55 54 4B 44 41 20 .BASS....UTKDA
00C0 20 20 20 2E 43 4D 44 53 07 00 0C 06 55 54 4B 44 .CMDS....UTKD
00D0 4F 20 20 20 20 2E 43 4D 44 53 07 00 0C 0D 55 43 O .CMDS....UC
00E0 4F 50 59 31 20 20 20 2E 42 41 53 53 04 00 0C 11 OPY1 .BASS....
00F0 55 43 4F 50 59 20 20 20 20 2E 43 4D 44 53 02 00 UCOPY .CMDS..
```

Note:

Le '?' à la fin sert pour afficher un dump du secteur.

Si on ne le met pas, le contenu sera directement affiché sans aucune transformation.

Fonctions spéciales:

Extraction d'un programme BASIC sous forme de fichier source

```
$ cp fs/AUTODOS\ .BAS? /tmp/AUTODOS.BAS
```

Rappel:

Le '?' à la fin permet de faire la traduction automatique

Transfert d'un programme source BASIC

Copie du programme texte source PGM.BAS avec conversion automatique :

```
$ cp /tmp/PGM.BAS fs/
```

L'extension '.BAS' est automatiquement reconnue et la conversion est faite à la volée!

Création d'un fichier '.tap'

Création d'un fichier '.tap' contenant le fichier AUTODOS\ .BAS

```
$ cp fs/AUTODOS\ .BAS?tape /tmp/autodos.tap
```

Vérification:

```
$ hexdump -vC /tmp/autodos.tap
00000000 16 16 16 24 00 00 00 00 05 c2 05 00 07 41 55 54 |...$......AUT|
00000010 4f 44 4f 53 00 00 17 05 01 00 c0 4d 41 53 54 45 |ODOS.....MASTE|
00000020 52 22 46 54 44 4f 53 33 2d 32 22 00 23 05 05 00 |R"FTDOS3-2".#...|
00000030 b9 23 34 38 43 2c 31 00 31 05 0a 00 8d 20 4a d4 |.#48C,1.1.... J.|
00000040 34 20 c3 20 36 00 3d 05 0f 00 b9 23 34 38 44 2c |4 . 6.=....#48D,|
00000050 4a 00 4c 05 14 00 8d 20 49 d4 31 20 c3 20 31 37 |J.L.... I.1 . 17|
00000060 00 58 05 1e 00 b9 23 34 38 45 2c 49 00 60 05 28 |.X....#48E,I.`.(|
00000070 00 c0 44 53 00 68 05 32 00 90 20 49 00 70 05 3c |..DS.h.2.. I.p.<|
00000080 00 90 20 4a 00 7c 05 46 00 b9 23 34 38 44 2c 4a |.. J.|.F..#48D,J|
00000090 00 8b 05 50 00 8d 20 49 d4 31 20 c3 20 31 34 00 |...P.. I.1 . 14.|
000000a0 97 05 5a 00 b9 23 34 38 45 2c 49 00 9f 05 64 00 |..Z..#48E,I...d.|
000000b0 c0 44 53 00 a7 05 6e 00 90 20 49 00 c1 05 78 00 |.DS...n.. I...x.|
000000c0 ba 22 46 54 44 4f 53 20 4d 49 53 20 45 4e 20 50 |."FTDOS MIS EN P|
000000d0 4c 41 43 45 22 00 00 00 |LACE"...|
000000d8
```

Remarque :

Le fichier généré n'est pas en démarrage automatique mais il est assez simple de modifier le plugin ou d'en faire un autre qui ne fasse que des '.tap' avec démarrage automatique.

Transfert du contenu d'un fichier '.tap'

L'extension '.tap' du fichier source permet la traduction directe :

```
$cp /tmp/obasefr.tap fs/
```

Après l'exécution de cette commande, la disquette contient les deux fichiers suivants:

```
-rw-rw-rw- 1 hcl users 13824 janv. 26 20:31 ORIC-BAS.BAS
-rw-rw-rw- 1 hcl users   256 janv. 26 20:31 obasefr.tap
```

Le premier est le programme BASIC contenu sur dans le fichier **obasefr.tap**
Le second contient des informations concernant le fichier **obasefr.tap**:

```
$ cat obasefr.tap
Fichier: ORIC BASE (ORIC BASE)
Start   : 0501
End     : 399A
Status  : AUTO (04)
Type    : Basic (00)
flag0   : 54 45
```

Avec le fichier **autodos.tap** précédent, on obtient:

```
-rw-rw-rw- 1 hcl users   512 janv. 26 20:36 AUTODOS-.BAS
-rw-rw-rw- 1 hcl users   256 janv. 26 20:36 autosdos.tap
```

```
$ cat autosdos.tap
Fichier: AUTODOS (AUTODOS )
Start   : 0500
End     : 05C2
Status  : STOP (00)
Type    : Basic (00)
flag0   : 00 00
```

Note :

Les fichiers '**xxxx.tap**' créés sur la disquette sont éphémères et ne sont pas conservés.
Il serait peut-être intéressant de générer automatiquement un fichier de type '**.TXT**' ou '**.SCR**' ou '**.HLP**' avec ces informations.

Affichage du contenu d'un fichier de données

```
$ cat ESSAI1\ \ .DAT?
#1
 1
 0
HELLO
-----
#2
 2
 501
HELLO
-----
#3
 3
 502
HELLO
-----
#4
 4
 503
HELLO
-----
#5
 5
 504
```

```

HELLO
-----
#6
 6
 505
HELLO
-----
#7
 7
 506
HELLO
-----
#8
 8
 507
HELLO
-----
#9
 9
 508
HELLO
-----
#10
 10
 509
HELLO
-----

```

La première ligne contient le numéro d'enregistrement, les lignes suivantes contiennent les champs de cet enregistrement.

Chaque enregistrement est séparé du suivant par une ligne '-----'

Fichiers ARY

Les fichiers de type '**.ARY**' ne supportent pas la conversion automatique pour l'affichage. Ceci est dû au fait que le type du tableau n'est pas stocké sur la disquette, il n'est donc pas possible de savoir si il s'agit d'entiers, de réels ou de chaînes.

Il faut préciser le type de données en ajoutant:

```

?read_array_int
?read_array_float
?read_array_string

```

Transfert d'un programme BASIC d'une image SEDORIC vers une image FT-DOS.

Exemple :

Copier le programme **ALPHA.COM** de la disquette SEDORIC vers une disquette FTDOS.

Disquette SEDORIC : **/tmp/SEDORIC.DSK**

Disquette FTDOS : **/tmp/FTDOS.DSK**


```
# Création des points de montage dans le répertoire courant
```

```
$ mkdir ftdos
```

```
$ mkdir sedoric
```

```
# Montage des images:
```

```
$ oricfs -o source=/tmp/SEDORIC.DSK ./sedoric
```

```
$ oricfs -o source=/tmp/FTDOS.DSK ./ftdos
```

```
# Copie du fichier, le '?' assure la conversion Token=>Texte et l'extension '.BAS' s'occupe de la conversion inverse
```

```
$ cp sedoric/ALPHA\ \ \ \ .COM? ftdos/ALPHA.BAS
```

ATTENTION:

Le programme ne fonctionnera pas si il utilise des commandes SEDORIC .

Format de fichier supportés

FTDOS

ARY

BAS

BIN

CMD

DAT

SCR

SYS

SEDORIC

COM

DAT

HLP

WIN

Utilitaires

Txt2bas / bas2txt

Ces deux utilitaires permettent de convertir un fichier texte source en un fichier basic binaire et réciproquement.

Syntaxe :

```
usage: txt2bas [-h] [--input INPUT] [--output OUTPUT] [--start START]
           [--version]
```

Convert BASIC source file to token file

optional arguments:

```
-h, --help                show this help message and exit
--input INPUT, -i INPUT   filename to convert (default: <open file '<stdin>',
                           mode 'r' at 0x7f53876b80c0>)
--output OUTPUT, -o OUTPUT tapename (default: <open file '<stdout>', mode 'w' at
                           0x7f53876b8150>)
--start START, -s START   start address (default: 1281)
--version, -v             show program's version number and exit
```

et :

```
usage: bas2txt [-h] [--input INPUT] [--output OUTPUT] [--start START]
           [--version]
```

Convert token file to BASIC source file

optional arguments:

```
-h, --help                show this help message and exit
--input INPUT, -i INPUT   filename to convert (default: <open file '<stdin>',
                           mode 'r' at 0x7f6e3997c0c0>)
--output OUTPUT, -o OUTPUT tapename (default: <open file '<stdout>', mode 'w' at
                           0x7f6e3997c150>)
--start START, -s START   start address (default: 1281)
--version, -v             show program's version number and exit
```

bin2tap

Cet utilitaire permet de transformer un fichier quelconque en fichier au format .tap.

```
usage: bin2tap [-h] [--input INPUT] [--output OUTPUT] [--start START] [--auto]
             [--fast] --type {basic,data,lscreen,hscreen} [--version]
```

Convert binary file to Oric .tap file

optional arguments:

```
-h, --help                show this help message and exit
--input INPUT, -i INPUT  filename to convert (default: <open file '<stdin>',
                        mode 'r' at 0x7f3bd41a70c0>)
--output OUTPUT, -o OUTPUT tapename (default: <open file '<stdout>', mode 'w' at
                        0x7f3bd41a7150>)
--start START, -s START  start address (default: None)
--auto, -a                autostart (default: False)
--fast, -f                fast speed tape (default: False)
--type {basic,data,lscreen,hscreen}, -t {basic,data,lscreen,hscreen}
                        file type (default: basic)
--version, -v            show program's version number and exit
```

L'indication du type de fichier doit être précisée.

Exemple :

```
$ bin2tap -i PILOT2.BIN -o pilot2.tap -t basic
```

```
Input: PILOT2.BIN, Output: pilot2.tap, TapeName: PILOT2
```

Annexes

Exemples de fichiers SCR

```
CAPS

!LOAD

Chargement d'un fichier BASIC ,
binaire ou de donnee.

SYNTAXE :

!LOAD "{NODSK.}NOMFICH{.EXT}{,DE}"

La valeur par defaut de NODSK est 1,
l'extension par defaut est BAS .
Lorsque le fichier est du type autre
que .BAS ,DE est l'adresse optionnelle
de zone memoire a partir de laquelle
on veut le charger .
```

Illustration 3: LOAD.SCR

```
CAPS

!CAT

Donne le Catalogue de la disquette
!CAT "" IMPRIME sur l'ecran
!LCAT "" IMPRIME sur l'imprimante

SYNTAXE : !CAT "2."
           pour obtenir le catalogue du
           disque 2
           !CAT "" donne le catalogue
           du disque 1
           Le disque 1. est pris par defaut

Pour lister sur imprimante votre cata-
logue faites: !LCAT"(NDISQ.)"
```

Illustration 2: CAT.SCR

Exemples de fichiers HLP

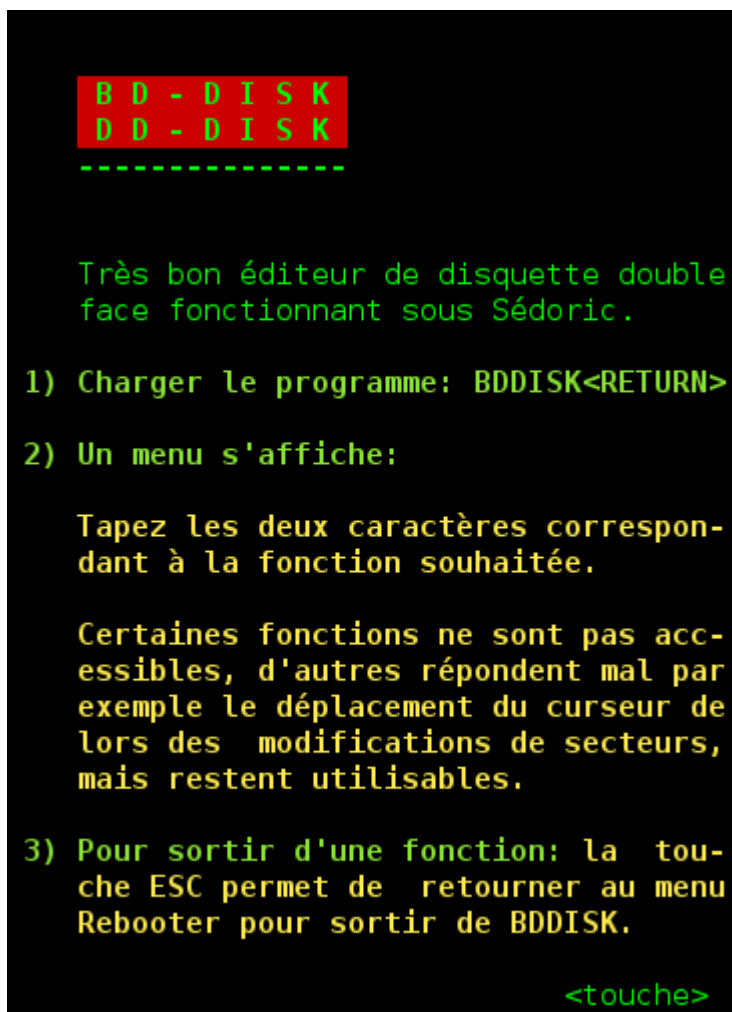


Illustration 4: BDDISK.HLP

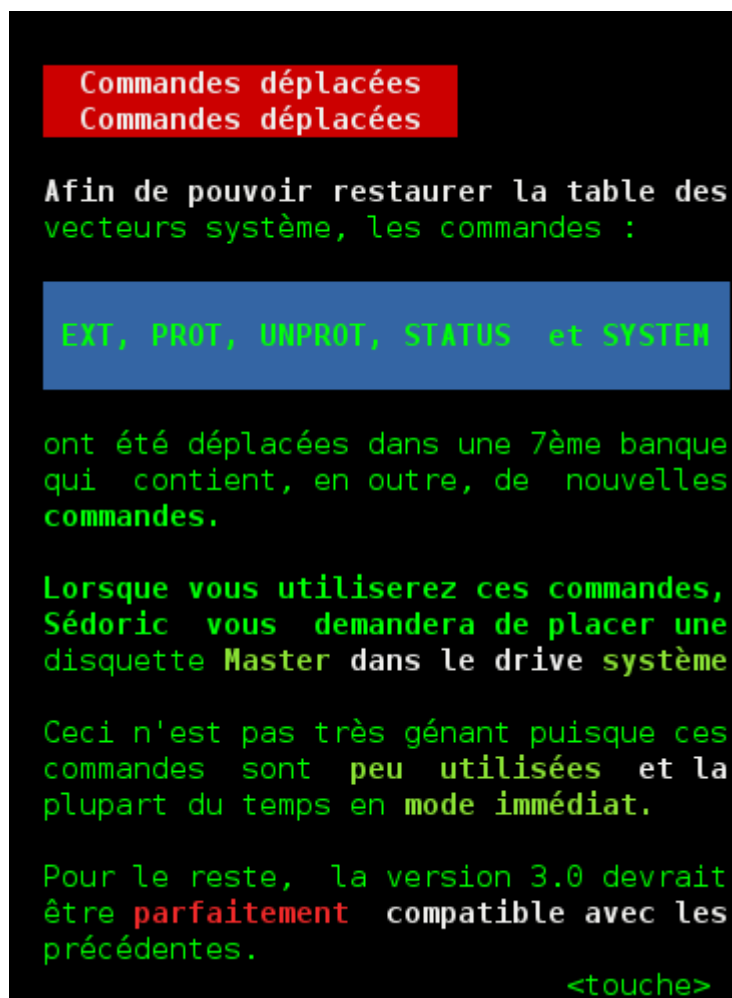


Illustration 5: V30NEWS02.HLP

```

Les touches de fonctions :
Les touches de fonctions :
-----> 1) FUNCT + touche
> 1) FUNCT + touche

A AZERTY B BACKUP C COPY D DIR
E ESAVE F FIELD G CHANGE H HCUR
I INIT J JUMP K KEYSAVE L LINPUT
M MOVE N NUM O OLD P PROT
Q QWERTY R RENUM S SAVEU T TYPE
U UNPROT V VIEWHIRES W WINDOW
X SEEK Y PAPER0:INK7 Z CALL#F8D0
1 DOKE#2F5,# vecteur ! et sa
2 DOKE#2F5,#467 valeur par défaut
3 DOKE#2F9,# vecteur ] et sa
4 DOKE#2F9,#D070 valeur par défaut
5 DOKE#2FC,# vecteur &() et sa
6 DOKE#2FC,#461 valeur par défaut
7 PAPER0:INK7 reset affichage
8 CALL#F8D0 reset caractères
9 ê 0 rien (un espace)
- ?HEX$(PEEK(#) = ?HEX$(DEEK(#)
_ PEEK(#) / DEEK(#)
[ POKE# ] DOKE#
<touche>

```

Illustration 6: V30NEWS05.HLP

Exemple de fichier WIN

```
FILE OF ADDRESSES
FILE OF ADDRESSES

INPUT / MODIFICATION OF A RECORD

Press: Ctrl-C to finish input, then
      <- or -> to move to next record
      S to save input, ESC for main menu

Name: 

Forename: 

Date of input:  /  / 19 

Address: 

Postal Code:  Town: 

Telephone number: 

Record: 
```

Illustration 7: ADDRESS.WIN