[LC565] TP C : Convertisseur BinHex

Introduction

Le format BinHex a été inventé pour répondre à deux besoins. D'une part celui de coder sur 7 bits seulement le contenu des fichiers binaires (donc codés sur 8 bits), afin de leur permettre de transiter par les réseaux, et par le courrier électronique, notamment. D'autre part celui de réunir les deux branches[1] des fichiers Macintosh en un seul fichier.

Enfin, l'auteur de cette norme, Yves Lempereur, a rajouté une fonction de compression élémentaire, de type RLE, pour éviter de transmettre des séquences de plusieurs caractères identiques.

Comment reconnaître un fichier codé en BinHex ?

L'extension utilisée pour les fichiers BinHex est généralement HQX, mais on rencontre aussi fréquemment des fichiers sans extension.

De plus, la première ligne (utile) des fichiers codés en BinHex est la suivante :

```
(This file must be converted with BinHex 4.0)
```

Il suffit donc d'ouvrir le fichier dans un programme de traitement de textes pour l'identifier. Cette ligne est normalement facultative, mais nous n'avons jamais rencontré de fichiers BinHex qui ne la comportent pas. L'une des caractéristiques de ces fichiers est également que le flux de données utile doit commencer et finir par le caractère ":".

En fait la ligne ci-dessus n'est pas forcément la première du fichier. Mais certains programmes ne reconnaissentpas le format si ce n'est pas la première. Il est donc conseillé, lorsqu'on écrit un fichier binhex de la mettre comme première ligne.

Quand peut-on rencontrer des fichiers BinHex ?

La norme BinHex est l'une des options (avec le format mime) de la plupart des logiciels de messagerie Macintosh. Ces programmes codent et décodent le format de manière transparente pour l'utilisateur. Toutefois, lorsque le courrier est échangé entre Mac et PC, il est possible que le logiciel de réception ne sache pas identifier ces fichiers et se contente de les stocker sur le disque dur.

Le format BinHex est également utilisé par certains programmes pour générer des fichiers Macintosh sur d'autres plates-formes.

---

[1] Les fichiers Macintosh sont constitués de deux parties appelées branches (forks):

DATA FORK:	The actual data included in the file. The Data fork is typically the only meaningful part of a Macintosh file on a non-Macintosh computer system. For example, if a Macintosh user wants to send a file of data to a user on an IBM-PC, she would only send the Data fork.
RESOURCE FORK:	Contains a collection of arbitrary attribute/value pairs, including program segments, icon bitmaps, and parametric values.

Appendix A. The BinHex format

Here is a description of the Hqx7 (7 bit format as implemented in BinHex 4.0) formats for Macintosh Application and File transfers.

The main features of the format are:

1) Error checking even using ASCII download
2) Compression of repetitive characters
3) 7 bit encoding for ASCII download

The format is processed at three different levels:

1) 8 bit encoding of the file:

```
Byte:    Length of FileName (1->63)
Bytes:   FileName ("Length" bytes)
Byte:    Version
Long:    Type                      (en fait 4 caractères ASCII)
Long:    Creator                   (idem)
Word:    Flags (And $F800)
Long:    Length of Data Fork
Long:    Length of Resource Fork
Word:    CRC
Bytes:   Data Fork ("Data Length" bytes)
Word:    CRC
Bytes:   Resource Fork ("Rsrc Length" bytes)
Word:    CRC
```

2) Compression of repetitive characters.

($90 is the marker, encoding is made for 3->255 characters)

```
00 11 22 33 44 55 66 77   -> 00 11 22 33 44 55 66 77
11 22 22 22 22 22 22 33   -> 11 22 90 06 33
11 22 90 33 44            -> 11 22 90 00 33 44
```

3) The whole file is considered as a stream of bits. This stream will be divided in blocks of 6 bits and then converted to one of 64 characters contained in a table. The characters in this table have been chosen for maximum noise protection. The format will start with a ":" (first character on a line) and end with a ":". There will be a maximum of 64 characters on a line. It must be preceded, by the following comment, starting in column 1 :

(This file must be converted with BinHex 4.0)

Any text before this comment is to be ignored.

The characters used is:

!"#$%&'()*+,-012345689@ABCDEFGHIJKLMNPQRSTUVXYZ[`abcdefhijklmpqr

11 90 90 90 90 90 —> 11 90 00 90 05.

(abcd) char a char b char c char d.

(char* abcd, char* abc)

Le TP proposé

Il s'agit d'écrire un convertisseur BinHex se limitant à la DATA FORK.

1) Ecrire une fonction qui :

* lit le fichier (supposé être un fichier texte) dont le chemin est donné en argument de la ligne de commande (attention ce chemin peut être une chaîne de caractères de longueur quelconque),
* vérifie que ce fichier contient une ligne avec le commentaire suivant (commençant à la colonne 1) :

  (This file must be converted with BinHex 4.0)
* décode le BinHex ce qui suit cette ligne,
* affiche les caractéristiques récupérées dans l'entête (nom, version, type, créateur, tailles des deux forks)
* place le DATA FORK dans un tableau d'octets de taille convenable.

On n'oubliera pas de faire les allocations mémoires qui s'imposent et de traiter les cas d'erreurs qui peuvent se rencontrer.

2) Ecrire une fonction qui sauve le DATA FORK précédemment créé sous le nom déterminé par l'entête, dans le même répertoire que le fichier décodé. Ici aussi pensez aux cas d'erreurs.

Vous pourrez alors tester votre programme sur les fichiers se terminant par **.hqx** qui se trouvent dans le répertoire **/users/tpinfo/blanc**. Ce sont des images que vous pourrez ensuite visualiser.

3) Complétez votre programme de façon à pouvoir faire la conversion inverse (en utilisant une option de la ligne de commande). On laissera vide le RESSOURCE FORK et on mettra le **type** correspondant à l'extension du fichier source suivant la table donnée en annexe ci-dessous, et le **créateur** UNIX. Appliquez cette conversion au(x) fichier(s) **gif**, **jpeg**... se trouvant dans le même répertoire **/users/tpinfo/blanc** que précédemment.