

Exemple d'entête JPEG

Cette annexe décortique un entête de fichier jpeg, c'est à dire tous les paramètres et tables décrits en [annexe A](#) qui viennent avant l'image encodée proprement dite. Conseil: regardez en parallèle ces deux annexes A et B!

On utilise les outils présentés en [section outils](#) et sur l'image fournie `poupoupidou_bw.jpg`. En version "pop art/pixel art monochrome" (élargie ici pour voir quelque chose) cette image ressemble à ça:



En version "sont fada ces profs de C, et dire que dans 3 semaines je lirai moi aussi dans la Matrice", la trace de `hexdump` est:

```
$ hexdump -C poupoupidou_bw.jpg

00000000  ff d8 ff e0 00 10 4a 46  49 46 00 01 01 00 00 01  |.....JFIF.....|
00000010  00 01 00 00 ff fe 00 10  3c 33 20 6c 65 20 70 72  |.....<3 le pr|
00000020  6f 6a 65 74 20 43 ff db  00 43 00 03 02 02 02 02  |ojet C...C.....|
00000030  02 03 02 02 02 03 03 03  03 04 06 04 04 04 04 04  |.....|
00000040  08 06 06 05 06 09 08 0a  0a 09 08 09 09 0a 0c 0f  |.....|
00000050  0c 0a 0b 0e 0b 09 09 0d  11 0d 0e 0f 10 10 11 10  |.....|
00000060  0a 0c 12 13 12 10 13 0f  10 10 10 ff c0 00 0b 08  |.....|
00000070  00 10 00 10 01 01 11 00  ff c4 00 15 00 01 01 00  |.....|
00000080  00 00 00 00 00 00 00 00  00 00 00 00 00 07 00 ff  |.....|
00000090  c4 00 22 10 00 02 02 01  04 02 03 01 00 00 00 00  |.."......|
000000a0  00 00 00 00 01 03 02 04  05 06 11 12 21 07 13 00  |.....!...|
000000b0  14 31 33 ff da 00 08 01  01 00 00 3f 00 47 f0 7b  |.13.....?.G.{|
000000c0  31 16 f4 d0 c1 d7 7b 31  56 68 29 a3 35 5d 22 0a  |1.....{1Vh).5]".|
000000d0  75 87 cf d6 c4 3b db ca  73 87 14 15 9d 87 0f eb  |u....;..s.....|
000000e0  d8 00 ec 6f 39 54 f1 ce  63 0f 7f 30 9c 72 9d a9  |...o9T..c..0.r..|
```

(et ça continue...)

Analyse section par section

Dans un élan de bienveillance infinie, voici la même chose mais en surlignant les marqueurs des différentes sections:

```
$ hexdump -C poupoupidou_bw.jpg

00000000 ff d8 ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 | .....JFIF.....|
00000010 00 01 00 00 ff fe 00 10 3c 33 20 6c 65 20 70 72 | .....<3 le pr|
00000020 6f 6a 65 74 20 43 ff db 00 43 00 03 02 02 02 02 | ojet C...C.....|
00000030 02 03 02 02 02 03 03 03 03 04 06 04 04 04 04 04 | .....|
00000040 08 06 06 05 06 09 08 0a 0a 09 08 09 09 0a 0c 0f | .....|
00000050 0c 0a 0b 0e 0b 09 09 0d 11 0d 0e 0f 10 10 11 10 | .....|
00000060 0a 0c 12 13 12 10 13 0f 10 10 10 ff c0 00 0b 08 | .....|
00000070 00 10 00 10 01 01 11 00 ff c4 00 15 00 01 01 00 | .....|
00000080 00 00 00 00 00 00 00 00 00 00 00 00 07 00 ff | .....|
00000090 c4 00 22 10 00 02 02 01 04 02 03 01 00 00 00 00 | ..".....|
000000a0 00 00 00 00 01 03 02 04 05 06 11 12 21 07 13 00 | .....!...|
000000b0 14 31 33 ff da 00 08 01 01 00 00 3f 00 47 f0 7b | .13.....?.G.{|
000000c0 31 16 f4 d0 c1 d7 7b 31 56 68 29 a3 35 5d 22 0a | 1.....{1Vh).5]".|
000000d0 75 87 cf d6 c4 3b db ca 73 87 14 15 9d 87 0f eb | u....;..s.....|
000000e0 d8 00 ec 6f 39 54 f1 ce 63 0f 7f 30 9c 72 9d a9 | ...o9T..c..0.r..|

(et ça continue...)
```

C'est déjà mieux non? Alors plus qu'à regarder en détail.

Prélude

Le premier marqueur est `ff d8`, c'est bien le *Start Of Image* (SOI).

APP0

Le second marqueur est `ff e0`, section *Application Data* (APP0) indiquant que l'image est de type JFIF. Les deux octets suivant le marqueur donnent la taille de la section, ici `00 10` (16) octets en comptant ces deux là. On peut vérifier que les cinq octets suivants sont bien `'J'`, `'F'`, `'I'`, `'F'`, `'\0'`, suivis de deux fois `'1'`, puis sept octets que l'on ignorera.

C'est conforme à la spécification vue en annexe [annexe A](#), donc on peut continuer.

Tout à fait Thierry

La prochaine section `ff fe` (COM) est un commentaire. La section est de taille 16 octets, donc vous trouverez quatorze octets à interpréter comme des caractères. Méditez ce commentaire.

Table de quantification

La section suivante `ff db` (DQT) définit une table de quantification. Sa taille est `00 43` soit 67 octets. Le premier octet après la taille est `00` : la précision est sur 8 bits et l'indice i_Q de la table est 0. Les 64 octets suivants `03 02 02 ...` sont les coefficients de cette table de quantification.

Ici on a bien lu la taille, un octet précision/indice et 64 valeurs. Ça fait 67 tout pile donc c'est fini. Tiens et si la taille de la section avait été plus grande? Allez, plus tard...

Start of Frame

La section suivante `ff c0` (SOF0) donne les informations relatives à l'image. Elle est de taille `00 0b` soit 11 octets. En continuant:

- le premier octet `08` indique une précision de 8 bits, ce qui est attendu en mode *baseline*.
- on trouve ensuite deux fois 2 octets pour la taille de l'image, ici 16x16 pixels.
- le prochain octet `01` indique qu'il n'y a qu'une seule composante, ce sera donc Y (monochrome).
- la table contient donc encore 3 octets: l'identifiant de la composante i_C (1), les facteurs d'échantillonnage (1x1), et l'indice de table de quantification associé à cette composante (0; ça tombe bien la table lue précédemment était d'indice $i_Q = 0$).

Table de Huffman

La prochaine section `ff c4` (DHT) définit une table de Huffman. La section est de taille `00 15` soit 21 octets. Le premier octet `00` indique une table pour une composante DC, d'indice 0. Les octets suivent le [format d'une table de Huffman JPEG](#):

- 16 octets donnant le nombre de codes de longueurs 1 à 16. Ici il n'y aura que deux codes de longueurs 1 et 2, respectivement.
- ensuite on lit les symboles eux-mêmes, ici 7 et 0.

Rebelote

Une autre section DHT? Pas de panique, vous savez la lire: AC, 15 symboles 1, 3, 2, 4, ..., 31, et 33.

SOS (*Start Of Scan*, pas Au Secours)

Ah voilà, la section suivante `ff da` (SOS) est celle qui va contenir les vraies données encodées. Elle est de taille 8, c'est bien $2N+6$ avec une seule composante.

Après la taille on trouve les octets `01 01 00 00 3f 00`. La spécification est super compliquée mais bon on retrouve qu'il n'y a bien qu'une seule composante dans ce scan et que son indice i_C est 1. Les deux derniers octets `3f 00` sont bien ceux attendus en mode *baseline*.

Et la suite? Et bien les octets restants `47 f0 7b ...` ne font plus partie de l'entête, c'est vraiment l'image encodée! Donc on sort du cadre de cette annexe, circulez.

This is The End

En réalité il reste encore un marqueur qu'on trouvera après avoir décodé l'image, c'est le `ff` `d9` (EOI) *End Of Image*.

Blablabla

Là nous venons de regarder le fichier binaire brut "à la dure" avec `hexdump`. Mais dans les outils nous vous avons aussi parlé de l'utilitaire `jpeg2blabla` codé avec soin et amour par nous-même. Il vous donnera les informations d'un entête de manière un peu plus interprétée, et aussi des détails sur les arbres de Huffman reconstruits.

En fait ces deux outils sont complémentaires, à vous de vous en servir à bon escient.

Et maintenant?

Cet exemple sur une image simple devrait être largement suffisant pour commencer. Ensuite ce sera à vous de savoir gérer les cas qui se présenteront pour des images plus générales.

Pour commencer ce projet, faites juste en sorte de lire les données d'entête correctement et de stocker les informations qui seront nécessaires pour la suite (même si vous ne les comprenez pas encore, ce qui viendra en grandissant).