

Musikbibliotek

De flesta formaten för ljud, bild och video innehåller så kallad metainformation, alltså information *om* informationen som filen i huvudsak innehåller. Det kan vara information om när och var en bild är tagen, hur långt ett filmklipp är, eller vilken artist som spelar en viss låt. Den här uppgiften går ut på att läsa av metainformationen hos ett antal musikfiler och sortera in dem i ett musikbibliotek.

Partiell lista över innehållet

- I/O från fil
- Minneshantering på bytenivå
- Interaktion med filsystemet

Introduktion

MPEG-1 Audio Layer 3 (ofta förkortat MP3) är ett filformat¹ för ljud som länge har varit populärt. Det finns ingen standard för metainformation i MP3-filer, men den överlägset mest använda metoden är att inkludera så kallade ID3-taggar. ID3 finns i flera versioner, men i den här uppgiften kommer vi fokusera på version 2.3 (som är den näst senaste).

ID3v2-taggar ligger ofta i början av ljudfilen och består av en *header* följt av ett antal *frames*. Headern består av 10 bytes och har följande utformning²:

```
ID3v2 identifier: "ID3"
ID3v2 version:  0x03 0x00
ID3v2 flags:    0x??
ID3v2 size:     0x?? 0x?? 0x?? 0x??
```

Först står de tre tecknen "ID3", vilket markerar starten på ID3-taggen. Sedan följer två bytes som anger vilken version av ID3v2 som används (här är det ID3v2.3.0). Påföljande byte anger ett antal flaggor som påverkar hur resten av filens metadata ska tolkas. Dessa kan ignoreras helt för denna uppgift. Slutligen följer fyra bytes som anger hur stor (i antal bytes) hela ID3-taggen är, exklusive headern. Se nedan för information om hur man tolkar dessa bytes!

Efter headern följer sedan ett antal poster (frames) som var och en innehåller ett visst stycke information. Varje post inleds med en egen header, också den 10 bytes, med följande utseende:

```
Frame ID:  0x?? 0x?? 0x?? 0x??
Size:      0x?? 0x?? 0x?? 0x??
Flags:     0x?? 0x??
```

Först står fyra tecken som anger vilken typ av information som posten innehåller (se nedan). Påföljande fyra bytes anger längden på posten (*exklusive* postens headern) på samma sätt som i ID3-taggens header. Sist ligger två bytes som anger flaggor om den aktuella posten. Dessa kan också ignoreras för den här uppgiften. Efter postens header kommer sedan postens själva innehåll, vars längd har angivits i postheadern.

Frameinnehåll

Vilken typ av innehåll som posten innehåller bestäms som sagt av de första fyra tecknen i postens header, som kallas Frame ID. Alla poster vars information består av text inleds med tecknet "T". I den här uppgiften kommer innehållet alltid vara text (se dock nedan om teckenkodning). Följande Frame ID:n är aktuella för den här uppgiften:

- TIT2 – Låttitel
- TPE1 – Huvudartist

¹Egentligen en algoritm för ljudkomprimering

²0x?? anger ett hexadecimalt tal (en byte).

- TALB – Album
- TRCK – Spårnummer
- TCON – Genre
- TYER – Inspelningsår

Padding

Efter den sista posten lägger man ibland in "tomt utrymme" bestående av bara nollor. Detta gör man för att lämna plats för fler frames innan filens ljuddata, så att man inte behöver skriva om hela filen om man vill utöka metainformationen. Notera att paddingen räknas in i längden som anges i ID3-taggens header!

ID3 size

Storleksangivelserna i ID3-formatet anges på ett lite speciellt sätt. Det är fyra bytes, där endast de sju *minst* signifikanta bitarna i varje byte spelar roll (den mest signifikanta biten sätts till noll). I praktiken är det alltså ett värde på 28 bitar. För att läsa in det till en int (till exempel) kan man läsa en byte i taget, "flytta över" de sju intressanta bitarna till inten och skjuta dem sju steg åt vänster för att lämna plats för bitarna i nästa byte. Följande bitoperationer kan vara användbara:

- $x \& k$ – Bitvis "och" mellan x och k . $x \& 127$ ger de sju minst signifikanta bitarna i x .
- $x \ll k$ – Skifta bitarna i x k steg åt vänster (och fyll på med nollor).

Teckenkodning

En teknisk detalj är att textinnehållet i en ID3-frame ibland använder en teckenkodning som inte är direkt kompatibel med ASCII. Det innebär att det kan vara svårt att läsa och använda textinformationen som den är skriven. För att ni ska slippa bry er om sådana lågnivådetaljer tillhandahåller vi ett bibliotek för att konvertera textinnehållet i en ID3-frame till UTF-8 (där alla tecknen i ASCII-tabellen kodas likadant som ASCII).

Genom att inkludera `convert_content.h` och kompilera programmet tillsammans med `convert_content.c` och flaggan `-liconv` (som länkar in biblioteket som används för kodningskonvertering) får ni tillgång till funktionen `convert_content`. Hur funktionen används står i `convert_content.h`.

Uppgiften

Målet med den här uppgiften är att skriva ett program som kan läsa in ID3-taggen från ett antal MP3-filer och skapa en katalogstruktur där filerna sorteras efter artist och album. Nedan följer ett antal tips för hur uppgiften kan angripas:

- Större delen av den här uppgiften går ut på att förstå hur ID3 fungerar och hur dess information skall läsas från filerna. Följande länkar kan vara av intresse:
 - Wikipedia:
<http://en.wikipedia.org/wiki/ID3>
 - ID3v2 made easy:
<http://id3lib.sourceforge.net/id3/easy.html>
 - ID3v2.3 informal standard:
<http://id3lib.sourceforge.net/id3/id3v2.3.0.html>
 - C file system interface:
http://www.gnu.org/software/libc/manual/html_node/File-System-Interface.html

- För att få ett hum om hur datan i MP3-filer ser ut kan ni prova att öppna en i Emacs. Det går att läsa en fil hexadecimalt genom att slå på läget `hexl-mode` (använd `M-x`).
- Ni kommer förmodligen vilja lagra information om individuella frames i ert program. Det kan vara en bra idé att skapa en strukt som kan lagra en frame i minnet.
- Försök inte skriva hela programmet på en gång! Se om ni först kan få programmet att skriva ut informationen i ID3-taggens header. Försök sedan skriva ut en fils alla frames.
- Prova funktionerna för interaktion med filsystemet i mindre program innan ni försöker bygga själva musikbiblioteket. När ni känner er säkra kan ni kan prova att först bara byta namn på filen ni läser (till något som anger lämplig metainformation), innan ni skapar några kataloger.

Musikfiler

I mappen `/it/kurs/imperoopmet/music` ligger ett antal filer i MP3-format. Ni kan använda dessa filer för när ni provkör ert program. De innehåller alla samma ljudinformation, men de har olika ID3-taggar. När ni ska testa att bygga upp ert musikbibliotek kan ni kopiera hela mappen till en katalog på ert konto och köra programmet.