

Karlheinz Brandenburg, director de tecnologías de medios electrónicos del Instituto Fraunhofer IIS, perteneciente al Fraunhofer-Gesellschaft —red de centros de investigación alemanes— que junto con Thomson Multimedia (renombrada como Technicolor) controlaba el grueso de las patentes relacionadas con el formato MP3. (Thomson inventó la tecnología lineal antifricción hace más de 60 años y ha seguido liderando la industria desde entonces. La marca Thomson está reconocida y comprobada como líder global en tecnología de movimiento mecánico.

En 2018, Thomson fue adquirido por Altra Industrial Motion; desde entonces, nuestra gama de productos ha crecido de manera significativa. Nuestra familia de productos para movimiento lineal y mecánico también incluye BSA, Neff, Tollo, Micron, Deltran y Cleveland: ahora todos forman parte de Thomson.)

Technicolor SA, anteriormente conocida como Thomson SA, y antes como Thomson Multimedia, es un fabricante francés multinacional de electrónica y proveedor de servicios multimedia establecido en Issy-les-Moulineaux, Francia. La compañía tiene al menos 60.000 empleados (2004), en 30 países.

El científico alemán Karlheinz Brandenburg es considerado como «el padre del MP3», y formó parte del equipo que le dio nombre al formato MPEG: «Grupo de Expertos de Imágenes en Movimiento» («Moving Pictures Experts Group»). El grupo cedió el nombre al método digital para comprimir señales de audio y video para facilitar su emisión y almacenamiento. El proceso llamado MPEG Audio Capa III (MPEG Audio Layer 3) es bastante común hoy en día pero su desarrollo no era fácil. Además otros soportes de audio ya usaban técnicas de compresión como el MPEG-1 Audio Layer I del casete compacto digital de Philips o el ATRAC usado por Sony para sus Minidisc.

## La historia inicia del mp3

En 1982, cuando Brandenburg participa en la creación del formato. Su tutor de tesis doctoral, había deseado patentar un método para transferir datos sin éxito. Lo que se pretendía patentar era una forma de transferir música usando líneas telefónicas, algo que la oficina de patentes germana consideraba imposible. No aparecería cierto progreso hasta 1986 cuando en la universidad de Ilmenau obtuvieron mejores computadoras para mejorar

la capacidad de trabajo. Se tuvieron que realizar esfuerzos para lograr el resultado deseado. Inicialmente se consideró usar un sistema por división de capas de sonido, pero fue desechado por ser considerado demasiado rígido, entonces se cambió por uno nuevo que aprovecha las limitaciones del oído humano.

La primera de ellas fue registrada en 1987, en ese año en el laboratorio de tecnologías de medios electrónicos, los alemanes intentaban resolver el dilema de como difundir el sonido digital. Los archivos en CD, eran pesados y engorrosos, las lectoras de CD eran novedad, también instalarlas en una PC.

En 1988 se convoca por parte de la Organización Internacional de Normalización al equipo de MPEG para crear un estándar de codificación de audio. En este momento se realizaban las pruebas del nuevo sistema. Parecía que finalmente lograron su objetivo, pero al momento de probarlo con «Tom's Dinner» de Suzanne Vega mostró fallas graves. Puesto que el formato de canción a capela y el escaso sonido ambiental representaron un enorme desafío para el sistema. Entonces se solicitó la colaboración de varias instituciones. Brandenburg trabajó con Jim Johnston de AT&T en el desarrollo de nuevos métodos de compresión para conservar la calidad de la voz. Finalmente lograron evitar que el sistema dañara la voz de la cantante, y tuvieron que trabajar más para finalmente lograr una calidad similar a la de un CD.

En 1992 la ISO incluyó al MP3 como un estándar de compresión de audio, pero no fue hasta el año siguiente cuando fue finalmente formalizado con la llegada del MPEG-1 Capa de Audio III (MPEG-1 Audio Layer III), con velocidades de muestreo de 33, 44,1 y 48 kHz. Entonces decidieron comercializarlo a empresas para transferir la música hacia los estudios de radio mediante RDSI.

Registraron varias patentes más en 1991, pero no fue hasta julio de 1995 cuando Brandenburg usó por primera vez la extensión `.mp3` para los archivos relacionados con el MP3 que guardaba en su ordenador. En el proceso de desarrollo del formato participó también el ingeniero Leonardo Chiariglione, quien tuvo la idea de los estándares que

podrían ser útiles para este fin.<sup>[5]</sup> Un año después su instituto ingresaba en concepto de patentes 1,2 millones de euros. Diez años más tarde esta cantidad ha alcanzado los 26,1 millones.

Entre 1994 y 1995 identificaron a la Internet como una área atractiva. Entonces decidieron darle finalmente a los archivos el nombre definitivo de mp3. El modelo de negocio se planeaba como herramientas de codificación costosas para empresas y decodificadores económicos para los consumidores. Uno de los productos decodificadores que lograron el mayor éxito y reconocimiento fue Winamp.

Sin embargo, no pasó mucho tiempo antes de que el formato fuera despojado del control de la ISO y Fraunhofer. Puesto que un ciudadano australiano había comprado el codificador usando una tarjeta de crédito robada de Taiwán para posteriormente empaquetarlo y cargarlo a un servidor FTP, de una universidad estadounidense. Se ignora hasta el momento la identidad y el paradero del infractor. Esto inició el conflicto entre la industria discográfica y el MP3, convirtiéndolo en su principal enemigo, aunque en Asia el éxito persistente de los discos compactos<sup>[6]</sup> esté ayudando a conservar las ventas en formato físico.

Tras el desarrollo de reproductores portátiles, y su integración en estéreos para automóviles, teléfonos móviles, reproductores de DVD, auriculares, consolas de videojuegos, altavoces y minisistemas de sonido hogareños, el formato MP3 en la actualidad llega más allá del mundo de la informática.

Soy un programador de C/C++ promedio. Recientemente tomé un proyecto para hacer un reproductor multimedia con una lista de reproducción inteligente que funcionará como el SmartDj de Zune. He decidido usar libvlc para jugar.

Nunca he codificado un software de código abierto antes, así que no sé nada sobre git y todo. ¿Me puede ayudar, por favor, a escribir al menos un programa en C para reproducir un archivo mp3?

¿Dónde debería empezar? ¿Cómo extraes el artista de una canción y otra información del propio archivo mp3?

Saludos.

Mejor respuesta

asegúrese de haber instalado los siguientes paquetes (de lo contrario, instálelo):

```
$ apt-get install libvlc-core dev libvlc-dev
```

prueba.c:

```
#include <stdio.h>

#include <stdlib.h>

#include <vlc/vlc.h>

int main(int argc, char **argv)
{
    libvlc_instance_t *inst;

    libvlc_media_player_t *mp;

    libvlc_media_t *m;

    // load the vlc engine

    inst = libvlc_new(0, NULL);

    // create a new item

    m = libvlc_media_new_path(inst, "path to MP3 file");

    // create a media play playing environment

    mp = libvlc_media_player_new_from_media(m);
```

```

// no need to keep the media now

libvlc_media_release(m);

// play the media_player

libvlc_media_player_play(mp);

sleep(10);

// stop playing

libvlc_media_player_stop(mp);

// free the media_player

libvlc_media_player_release(mp);

libvlc_release(inst);

return 0;
}

```

como enlazar y compilar:

```

$ gcc $(pkg-config --cflags libvlc) -c test.c -o test.o

$ gcc test.o -o test $(pkg-config --libs libvlc)

```

imero de todo, escribir el siguiente código:

```
#include <Mmsystem.h>
```

```
#include <mciapi.h>
```

```
//these two headers are already included in the <Windows.h> header
```

```
#pragma comment(lib, "Winmm.lib")
```

Para abrir \*.mp3:

```
mciSendString("open \"*.mp3\" type mpegvideo alias mp3", NULL, 0, NULL);
```

Jugar \*.mp3:

```
mciSendString("play mp3", NULL, 0, NULL);
```

A play y esperar hasta que el \*.mp3 ha terminado de jugar:

```
mciSendString("play mp3 wait", NULL, 0, NULL);
```

A replay (volver a jugar desde el inicio)\*.mp3:

```
mciSendString("play mp3 from 0", NULL, 0, NULL);
```

Para repetir y esperar hasta que el \*.mp3 ha terminado de jugar:

```
mciSendString("play mp3 from 0 wait", NULL, 0, NULL);
```

Para jugar el \*.mp3 y reproducción cada vez que termina como un bucle:

```
mciSendString("play mp3 repeat", NULL, 0, NULL);
```

Si quieres hacer algo cuando el \*.mp3 ha terminado de jugar, entonces usted necesita para `RegisterClassEx` por el `WNDCLASSEX` estructura, `CreateWindowEx` y proceso de los mensajes con el **GetMessage**, **TranslateMessage** y **DispatchMessage** funciones en un `while` bucle y llamada:

```
mciSendString("play mp3 notify", NULL, 0, hwnd); //hwnd is an handle to the window returned from CreateWindowEx.
```

```
If this doesn't work, then replace the hwnd with MAKELONG(hwnd, 0).
```

En el procedimiento de ventana, añadir el `case MM_MCINOTIFY:` El código no se ejecutará cuando el mp3 ha terminado de jugar.

**Pero** si usted programa una **Consola** de la Aplicación y no tiene que lidiar con **windows**, entonces usted puede `CreateThread` en estado de suspensión especificando la `CREATE_SUSPENDED` bandera en el `dwCreationFlags` parámetro y mantener el valor de retorno en una `static` variable y llamarlo lo que quieras. Por ejemplo, yo me llamo mp3. El tipo de este `static` variable es `HANDLE` de curso.

Aquí es el `ThreadProc` para la `lpStartAddress` de este hilo:

DWORD WINAPI MP3Proc(\_In\_ LPVOID lpParameter) //lpParameter can be a pointer to a structure that store data that you cannot access outside of this function. You can prepare this structure before `CreateThread` and give it's address in the `lpParameter`

```
{
```

```
Data *data = (Data*)lpParameter; //If you call this structure Data, but you can call it whatever you want.
```

```
while (true)
```

```
{
```

```
mciSendString("play mp3 from 0 wait", NULL, 0, NULL);
```

```
//Do here what you want to do when the mp3 playback is over
```

```
SuspendThread(GetCurrentThread()); //or the handle of this thread that you keep in a static variable instead
```

```
}
```

```
}
```

Todo lo que tienes que hacer ahora es `ResumeThread(mp3)`; cada vez que se desea reproducir tus mp3 y algo va a suceder cada vez que se termina.

Puede `#define play_my_mp3 ResumeThread(mp3)`; para hacer el código más legible.

Por supuesto, usted puede quitar el `while (true)`, `SuspendThread` y la **de 0** códigos, si quieres jugar con tu archivo mp3 sólo **una vez** y hacer lo que quiera cuando se termina.

Si **sólo** quitar el `SuspendThread` llamada, el sonido se reproducirá una y otra vez y hacer algo siempre es más. Esto es equivalente a:

```
mciSendString("play mp3 repeat notify", NULL, 0, hwnd); //or MAKELONG(hwnd, 0) instead
```

en windows.

Para pausar el \*.mp3 de en medio:

```
mciSendString("pause mp3", NULL, 0, NULL);
```

y reanudar:

```
mciSendString("resume mp3", NULL, 0, NULL);
```

A parar en medio:

```
mciSendString("stop mp3", NULL, 0, NULL);
```

Tenga en cuenta que no se puede reanudar un sonido que ha sido detenido, pero sólo hizo una pausa, pero puede reproducir mediante la realización de la **jugar** comando.

Cuando termines de jugar este \*.mp3, no te olvides de:

```
mciSendString("close mp3", NULL, 0, NULL);
```

Todas estas acciones también se aplican a (trabajar con) archivos de la onda, pero con la onda de los archivos, puede utilizar «waveaudio» en lugar de «mpegvideo».

También puedes jugar con ellos directamente, sin necesidad de abrirlos:

```
PlaySound("*.wav", GetModuleHandle(NULL), SND_FILENAME);
```

Si usted no desea especificar un identificador de módulo:

```
sndPlaySound("*.wav", SND_FILENAME);
```

Si usted no quiere esperar hasta que la reproducción es más:

```
PlaySound("*.wav", GetModuleHandle(NULL), SND_FILENAME | SND_ASYNC);
```

```
//or
```

```
sndPlaySound("*.wav", SND_FILENAME | SND_ASYNC);
```

Para reproducir el archivo de onda, una y otra vez:

```
PlaySound("*.wav", GetModuleHandle(NULL), SND_FILENAME | SND_ASYNC | SND_LOOP);
```

```
//or
```

```
sndPlaySound("*.wav", SND_FILENAME | SND_ASYNC | SND_LOOP);
```

Tenga en cuenta que debe especificar tanto el `SND_ASYNC` y `SND_LOOP` banderas, porque nunca se va a esperar hasta que un sonido, que se repite infinidad de veces, es lo más!

También puede `fopen` el archivo de onda y copiar todos sus bytes a un buffer (una enorme/enorme (muy grande) de la matriz de bytes) con el `fread` función y, a continuación:

```
PlaySound(buffer, GetModuleHandle(NULL), SND_MEMORY);
```

```
//or
```

```
PlaySound(buffer, GetModuleHandle(NULL), SND_MEMORY | SND_ASYNC);
```

```
//or
```

```
PlaySound(buffer, GetModuleHandle(NULL), SND_MEMORY | SND_ASYNC | SND_LOOP);
```

```
//or
```



```
sndPlaySound(buffer, SND_MEMORY);
```

```
//or
```

```
sndPlaySound(buffer, SND_MEMORY | SND_ASYNC);
```

```
//or
```

```
sndPlaySound(buffer, SND_MEMORY | SND_ASYNC | SND_LOOP);
```

Ya sea `OpenFile` o `CreateFile` o `CreateFile2` y bien `ReadFile` o `ReadFileEx` funciones se pueden utilizar en lugar de `fopen` y `fread` funciones.