

## Multimídia multiplataforma

# SDL e SMPEG

Neste artigo você irá aprender um pouco sobre as bibliotecas SDL e SMPEG e sobre o formato MPEG de compactação de vídeo e áudio. Além disso, você irá interagir com estas maravilhas do mundo multimídia escrevendo um pequeno aplicativo em C para tocar vídeos MPEG.



POR DIEGO FIORI DE CARVALHO E RENAN PRATES LOPES DE CAMPOS

A API livre SDL, Simple DirectMedia Layer, é uma ferramenta multi-plataforma para o desenvolvimento de aplicações multimídia e bastante utilizada para o desenvolvimento de jogos para a plataforma Linux. Se você procura uma alternativa Linux ao DirectX, a SDL é uma boa escolha.

Essa biblioteca permite o acesso a vários recursos de hardware, como áudio, mouse, joystick e teclado. É utilizada para o desenvolvimento de players de vídeo, emuladores, jogos e outros aplicativos multimídia. A SDL ainda possui suporte a vários sistemas operacionais, entre eles o Linux, Windows, BeOS, MacOS Classic, MacOS X, FreeBSD, OpenBSD, BSD/OS, Solaris, IRIX, e QNX.

Os recursos oferecidos permitem que sejam desenvolvidos diversos programas de modo eficiente. Ou seja, onde os recursos multimídia sejam explorados de maneira fácil e com bom desempenho. Essa flexibilidade da SDL, e o fato da biblioteca estar sob licença LGPL, contribuiu para o seu contínuo desenvolvimento e agrega um número cada vez maior de usuários. É importante ressaltar que, mesmo sendo uma API livre, no Windows a SDL é dependente da DirectX.

## SDL MPEG Player Library

A SDL MPEG Player Library (SMPEG) é uma biblioteca de funções para exibição de vídeos no formato MPEG com suporte a áudio, desenvolvida por Woo-jae Jung. Ela é baseada no Berkeley MPEG player, e está sob a licença LGPL. Suporta a reprodução de vídeos no formato MPEG-1, além do suporte a

áudio no formato MP3. Sua implementação visa o uso mínimo de recursos do sistema, porém seu desempenho deixa a desejar em processadores inferiores a um Pentium II.

A SMPEG é baseada na SDL, e necessita dela para funcionar. É a partir dos recursos básicos da SDL que são implementados os métodos de exibição de vídeos, que permitem a decodificação e exibição dos filmes MPEG com suporte a áudio. Utilizando os recursos de ambas as bibliotecas é que são gerados os programas que exibem, de fato, os vídeos MPEG.

## MPEG

*Moving Pictures Experts Group* (MPEG) é o nome do comitê ISO que trabalha com compressão de áudio e vídeo digital. A especificação MPEG-1 é desenvolvida especialmente para permitir a transmissão de imagens em qualidade similar à de uma fita VHS a uma taxa de dados entre 1 e 1.5 Mbps.

O padrão MPEG-1 divide-se em codificação de vídeo, codificação de áudio e sistemas. A subdivisão de sistemas inclui informações necessárias para a sincronia entre as streams de áudio e vídeo. A stream de vídeo utiliza aproximadamente 1.15 Megabits (Cerca de 147 KBytes) por segundo, e o restante da largura de banda é usado pelas stream de áudio e dados do sistema (aproximadamente 0.35 Megabits por segundo).

As imagens em MPEG são formatadas por três matrizes retangulares de inteiros: uma de luminância (Y) e duas de crominância (Cb e Cr), esse padrão YCrCb refere-se à uma forma de repre-

sentação de cores bastante usada em compressão de imagens.

O algoritmo para áudio MPEG é bem complexo, formado por um conjunto de três sistemas de codificação e decodificação, chamadas de Camada 1 (*Layer I* menos complexa), Camada 2 (*Layer II*) e Camada 3 (*Layer III* mais complexa). Essas camadas são hierarquicamente compatíveis, permitindo que um decodificador da Camada 3 trabalhe com dados comprimidos usando os métodos das Camadas 1 e 2, por exemplo.

Camada 1: Esta permite a construção de simples codificadores e decodificadores de áudio com um desempenho médio e que operam razoavelmente bem a 192 ou 256 Kbps por canal.

Camada 2: É otimizada para um bitrate de 96 ou 128 Kbps por canal monofônico. No modo estereofônico, este sistema MPEG pode ter qualidade equivalente à qualidade de um CD.

Camada 3: sistema de maior desempenho na série, é basicamente um esquema de compressão de som, objetivando um bitrate de 64 Kbps por canal de áudio. É o famoso MP3, oficialmente chamado *MPEG-1 Layer III audio*.

As especificações do padrão MPEG têm sido divididas pelo grupo ISO em algumas fases. Essas fases não ocorrem rigorosamente em uma ordem consecutiva, havendo sobreposição entre algumas delas. As fases são: MPEG-1, MPEG-2, MPEG-3, MPEG-4. Vamos exemplificar somente a primeira fase, pois é esta com a qual a biblioteca SMPEG trabalha.

MPEG-1 destina-se ao armazenamento de seqüências de vídeo a cores com áudio sincronizado, com qua-

lidade similar à de um VCR sobre CD-ROMs. Essa opção tem um bitrate de 1.2 Mbps. Na prática, se o canal de áudio exigir 200 a 250 Kbps (para som em qualidade de CD), o bitrate do canal de vídeo permanecerá entre 1.15 a 1.2 Mbps. A razão de compressão de vídeo é na ordem de 26:1 [1]. Seu principal uso é no antecessor do DVD, um formato conhecido como VCD, muito popular na Ásia. [2]

## Desenvolvendo o aplicativo

Hoje em dia a maioria das distribuições já incluem o pacote SDL na instalação padrão. Caso precise instalá-la manualmente, vá até o site oficial [3], e baixe a última versão estável. A SMPEG pode ser encontrada em [4]. A instalação a partir do código-fonte segue o procedimento padrão, com os comandos: `./configure ; make ; make install`.

No exemplo mostrado na listagem 1 criamos um programa em C (mesma linguagem de programação usada para escrever a SDL) que exibe um arquivo

MPEG e permite ao usuário parar, pausar e capturar um screenshot do vídeo. As instruções de uso são simples: tecla *Espaço* para pausar/reproduzir o vídeo, *ESC* para sair e *P* para capturar um screenshot no formato BMP. O programa exibe o vídeo em tela cheia e pode ser compilado e executado tanto no Linux como Windows. Para compilá-lo no Linux use o comando:

```
gcc -I/usr/include/SDL -I/usr/
include/smpeg -lSDL -lsmpeg -o
sdl_smpeg sdl_smpeg.c
```

## No final das contas...

Essa aplicação é apenas uma demonstração do que pode ser feito com o uso dessas bibliotecas. A SDL suporta programação com multi-threading, além de possuir muitos eventos para programação de aplicações relacionadas a áudio, vídeo, além de funções para manipulação de dispositivos de Entrada/Saída, tais como mouse e CDRoms. Entre outras coisas ela permite que sejam

criados jogos, com a leitura de eventos de joysticks e vários outros dispositivos de entrada, e muito mais. ■

## INFORMAÇÕES

- [1] Rudinei Goularte e Edson dos Santos Moreira, "Produção de Vídeo Digital Teoria e Prática" – Notas didáticas do ICMC/USP, Fevereiro, 1998.
- [2] VCD: [http://en.wikipedia.org/wiki/Video\\_CD](http://en.wikipedia.org/wiki/Video_CD)
- [3] SDL: <http://www.libsdl.org>
- [4] SMPEG: <http://www.lokigames.com/development/smpeg.php3>
- [5] SDL Tutorial: <http://www.libsdl.org/tutorials.php>
- [6] SMPLAY, projeto de Woo Jae Jung: <http://splay.sourceforge.net>
- [7] Introduction on SDL--Video & OpenGL with SDL: <http://sdl.doc.csn.ul.ie/guidevideo.php>
- [8] VideoHelp: <http://www.videohelp.com/vcd>

## Programa : sdl\_smpeg.c

```
001 #include <stdio.h> // que utilizam vídeos em vez de animações. Você
002 #include <stdlib.h> // simplesmente trocará o vídeo sem a
003 // necessidade de criar outra janela.
004 #ifdef unix 019
005     #include <SDL/SDL.h> 020 SMPEG * inicializa_smpeg (char * arquivo_video, 2
006     #include <SDL/SDL_syswm.h> SDL_Surface * tela);
007     #include <smpeg/smpeg.h> 021
008 #else 022 SDL_Surface * inicializa_sdl (char * arquivo_2
009 #ifdef WIN32 023
010     #include "SDL.h" 024 // O programa principal basicamente inicia a
011     #include "smpeg.h" 025 // exibição do vídeo e aguarda pela ocorrência
012     #include <windows.h> 026 // de um evento de mouse ou teclado.
013 #endif // WIN32 027
014 #endif // unix 028 {
015 029 // Essa é uma estrutura de dados da
016 // Logo abaixo estão as prototipações das 030 // biblioteca SMPEG que faz referência
// funções que inicializam a SDL e a SMPEG 031 // à um arquivo de vídeo no formato
// respectivamente. // MPEG.
017 // A inicialização da SDL consiste na criação de 029 SMPEG *mpeg;
// uma janela (superfície) onde será exibido o 030
// vídeo. A inicialização da SMPEG define qual o 031
// arquivo de vídeo que será exibido e em que // Essa é uma estrutura de dados da
// superfície da SDL. // biblioteca SDL que armazena conjuntos
018 // Então por que não juntar as duas em uma // de pixels. É por meio dela que é
// única função? Pode ser interessante que na // possível desenhar, plotar e extrair
// ocorrência de um evento, por exemplo um // figuras de um aplicativo que utiliza a
// clique de mouse, você queira trocar // SDL. Nesse exemplo essa estrutura
// rapidamente o vídeo que está sendo exibido // será usada para exibir o nosso vídeo.
// por outro, como naqueles jogos de tiroteio
```

## Programa : sdl\_smpeg.c

```

032     SDL_Surface *tela;                                084         case SDL_MOUSEBUTTONDOWN:
033                                                     085
034     // Estrutura que contem informações do           086         // A coordenada do
// vídeo.                                           // clique e o frame do
035     SMPEG_Info info;                                  // vídeo são impressos
036                                                     // no terminal.
037     SDL_Event evento;                                087         // Caso o mouse tenha
038                                                     // dois ou três botões,
039     int sair,                                         // é possível tratá-los
040         frame,                                       // separadamente.
041         x,                                           088         SMPEG_getinfo (mpeg, &
042         y,                                           &info);
043         pause;                                       089
044                                                     090         x = (int) (evento.button
045     float razao_x,                                   x * razao_x);
046         razao_y;                                       091         y = (int) (evento.button
047                                                     y * razao_y);
048     mpeg = NULL;                                       092
049     tela = NULL;                                       093         frame = info.current_
050         frame;                                       frame;
051     sair = 0;                                           094
052     pause = 0;                                       095         printf ("%d %d %d\n",
053         frame, x, y);
054     razao_x = 1;                                       096
055     razao_y = 1;                                       097         break;
056                                                     098
057     if (argc == 1 || argc > 2)                       099         // Caso o evento seja uma tecla
058     {                                                 // pressionada...
059         printf ("\nParametro incorreto.\n");        100         case SDL_KEYDOWN:
060         exit (1);                                       101
061     }                                                 102         // Se a tecla for "ESC"
062                                                     // então saia do loop.
063     if ((tela = inicializa_sdl (argv[1], &          103         if (evento.key.keysym.sym
razao_x, razao_y)) == NULL)                          == SDLK_ESCAPE)
064     {                                                 sair = 1;
065         exit (1);                                       105
066     }                                                 106
067                                                     // Se a tecla for
068     if ((mpeg = inicializa_smpeg (argv[1], &        // "espaço" então toque
tela)) == NULL)                                       // ou pause o video.
069     {                                                 else if (evento.key.keysym
070         exit (1);                                       sym == SDLK_SPACE)
071     }                                                 {
072                                                     if (pause)
073     SMPEG_play (mpeg);                                       {
074         play (mpeg);                                       SMPEG_
075     // Enquanto sair for falso faça...                pause
076     while (sair == 0)                                       = 0;
077     {                                                 }
078         // SDL aguarda por um evento.                else
079         SDL_WaitEvent (&evento);                       {
080                                                     SMPEG_
081         switch (evento.type)                               pause (mpeg);
082         {                                                 pause
083             // Caso o evento seja um clique          = 1;
// do mouse...                                       }

```

## Programa : sdl\_smpeg.c

```

119         }
120
121         // Se a tecla for "p"
122         // então capture um
123         // screenshot.
124         v
125         else if (evento.key.↵
126         keysym.sym == SDLK_p)
127         {
128             SDL_LockSurface↵
129             (tela);
130             SDL_SaveBMP ↵
131             (tela, "screenshot.bmp");
132             SDL_Unlock↵
133             Surface (tela);
134             SDL_Flip (tela);
135         }
136         break;
137
138         case SDL_QUIT:
139             sair = 1;
140             break;
141         default:
142             break;
143     }
144
145     // Se o vídeo acabou...
146     if (SMPEG_status (mpeg) ↵
147     == SMPEG_STOPPED && pause == 0)
148         // Saia do loop.
149         sair = 1;
150     }
151
152     // Finalização do vídeo:
153     SMPEG_stop (mpeg);
154
155     // Libera o ponteiro
156     SMPEG_delete (mpeg);
157
158     // Finalização da SDL:
159     SDL_Quit ();
160
161     return (0);
162 }
163
164 SMPEG *
165 inicializa_smpeg (char * arquivo_video, SDL_↵
166 Surface * tela)
167 {
168     // Nosso arquivo de vídeo.
169     SMPEG * mpeg = NULL;
170
171     // O ponteiro SMPEG é apontado para o
172     // arquivo de vídeo passado como
173     // parâmetro do programa.
174     if ((mpeg = SMPEG_new (arquivo_video, ↵
175     NULL, 1)) == NULL)
176     {
177         return NULL;
178     }
179
180     // Esta função define em qual superfície
181     // da SDL o vídeo será exibido.
182     SMPEG_setdisplay (mpeg, tela, NULL, ↵
183     NULL);
184
185     if (SMPEG_error (mpeg) != NULL)
186     {
187         return NULL;
188     }
189
190     // O parâmetro '1' habilita a saída de ↵
191     vídeo.
192     SMPEG_enablevideo (mpeg, 1);
193
194     if (SMPEG_error (mpeg) != NULL)
195     {
196         return NULL;
197     }
198
199     // O parâmetro '1' habilita a saída
200     // de áudio.
201     SMPEG_enableaudio (mpeg, 1);
202
203     if (SMPEG_error (mpeg) != NULL)
204     {
205         return NULL;
206     }
207
208     return mpeg;
209 }
210
211 SDL_Surface *
212 inicializa_sdl (char * arquivo_video, int ↵
213 razao_x, int razao_y)
214 {
215     // Nossa janela de exibição.
216     SDL_Surface * tela = NULL;
217
218     // É nessa função que é definido o
219     // tamanho da janela da SDL. Entao o
220     // cálculo da razao entre as resoluções
221     // do vídeo com a do sistema deve ser
222     // feita aqui. Então usamos a estrutura
223     // da SMPEG para abrir o arquivo de
224     // vídeo e conhecer sua resolução.
225     SMPEG * video = NULL;
226
227     // Essa é a estrutura que contém a
228     // resolução do arquivo de video.
229     SMPEG_Info info;
230
231     // Essa é uma estrutura de dados da

```

## Programa : sdl\_smpeg.c

```

// biblioteca SDL que armazena
// informações do gerenciador de janelas
// do sistema. No caso do Linux, o
// gerenciador de janelas é o X11.
208 SDL_SysWMInfo wmInfo;
209
210 // Essa é uma estrutura de dados da
// biblioteca SDL que armazena
// informações de vídeo do sistema.
// Nesse exemplo, a informação desejada
// é o número bits per pixel (bpp) do
// gerenciador de janelas.
211 const SDL_VideoInfo * videoInfo;
212
213 int w, // Resolução horizontal.
214 h, // Resolução vertical.
215 bpp; // Bits per pixel
216
217 // Inicializa o sistema de vídeo da SDL.
218 if (SDL_Init (SDL_INIT_VIDEO) < 0)
219 {
220     return NULL;
221 }
222
223 // Inicializa o sistema de áudio da SDL.
224 if (SDL_Init (SDL_INIT_AUDIO) < 0)
225 {
226     return NULL;
227 }
228
229 if ((video = SMPEG_new (arquivo_video, &info, 0)) == NULL)
230 {
231     return NULL;
232 }
233
234 SDL_VERSION (&wmInfo.version);
235
236 if (SDL_GetWMInfo (&wmInfo) == 0)
237 {
238     return NULL;
239 }
240
241 // Poderíamos inicializar a superfície
// da SDL com um tamanho pré-definido,
// mas é mais interessante ajustar a
// superfície para tela cheia.
242 // No Linux...
243 #ifdef unix
244 // Se o gerenciador de janelas é o X11
245
246 if ( wmInfo.subsystem == SDL_SYSWM_X11 )
247 {
248     wmInfo.info.x11.lock_func();
249     w = DisplayWidth (wmInfo.info.x11.display, zDefaultScreen (wmInfo.info.x11.display));
250     h = DisplayHeight (wmInfo.info.x11.display, DefaultScreen (wmInfo.info.x11.display));
251     wmInfo.info.x11.unlock_func();
252 }
253 // Se não estiver no X11...
254 else
255 {
256     return NULL;
257 }
258 #else
259 #ifdef WIN32
260 // No caso do Windows...
261 BOOL rv;
262 RECT rect;
263 rv = SystemParametersInfo (SPI_GETWORKAREA, sizeof (RECT), &rect, 0);
264 w = (int)(rect.right - rect.left);
265 h = (int)(rect.bottom - rect.top);
266 #endif // WIN32
267 #endif // unix
268 // As razões são usadas para ajustar
// as coordenadas dos cliques do mouse
// sobre a superfície SDL de acordo com
// a resolução do arquivo de vídeo MPEG.
// Desse modo, se o arquivo de vídeo
// possui 640x480 pixels, a coordenada
// do clique do mouse será dada em razão
// dessa resolução.
269 razao_x = (float) info.width / w;
270 razao_y = (float) info.height / h;
271
272 SMPEG_delete(video);
273 if ((videoInfo = SDL_GetVideoInfo()) == NULL)
274 {
275     return NULL;
276 }
277 bpp = videoInfo->vfmt->BitsPerPixel;
278
279 // Os parâmetros determinam,
// respectivamente, que a janela não
// exibirá decorações, será armazenada
// na memória de vídeo e exibida
// em tela-cheia.
280 if ((tela = SDL_SetVideoMode (w, h, bpp, SDL_NOFRAME | SDL_HWSURFACE | SDL_FULLSCREEN)) == NULL)
281 {
282     return NULL;
283 }
284 return tela;
285 }

```