



## 12. Cavalos de Troia em ANSI-C

---

Dr. Sidney Marlon Lopes de Lima

Currículo *Lattes*: <http://lattes.cnpq.br/0323190806293435>

Departamento de Eletrônica e Sistemas - Universidade Federal de Pernambuco, Recife, Pernambuco. [sidney.lima@ufpe.br](mailto:sidney.lima@ufpe.br)

### 12.1 Introdução

Até o momento, os experimentos práticos foram desenvolvidos em linguagem *Python*. Apesar de toda praticidade para desenvolvimento, um programa em *Python* não é executado diretamente no Sistema Operacional. As instruções em *Python* são executadas na máquina virtual. Por sua vez na máquina virtual, há a interpretação das instruções (comandos) *Python* e e sua conversão para a(s) instrução(ões) equivalente(s) do Sistema Operacional. Como o aplicativo não é executado no Sistema Operacional, conseqüentemente depende-se que a máquina virtual *Python* esteja previamente instalada na máquina real. Em termos práticos, uma vítima só seria infectada por um aplicativo *malware Python* se houvesse a instalação prévia de uma máquina virtual *Python*. Por essa razão, as *botnet*<sup>1</sup> e cavalos de troias são desenvolvidas em linguagens compiladas em detrimento de *Python*. Em termos coloquiais; ANSI-C é para atacar, *Python* é para defender.

Um aplicativo, gerado por uma linguagem compilada, não usa máquina virtual. Um *malware* compilado é executado diretamente no Sistema Operacional (e.g: Windows 7). Alguns exemplos de linguagens compiladas são ANSI-C, C++ e C#. O Gerenciador de tarefas mostra os processos e os serviços que estão sendo executados no computador. Para acessar o Gerenciador de tarefas, basta clicar com o lado direito do mouse na barra de tarefas. O aplicativo compilado quando executado diretamente no Sistema Operacional pode ser visto no Gerenciador de Tarefas conforme Fig. 12.2.

No presente Capítulo, será aplicado o compilador gcc<sup>2</sup> responsável por gerar aplicativos executáveis a partir de códigos-fontes em linguagem ANSI-C. Como ambiente de desenvolvimento,

---

<sup>1</sup>Botnet: releia a definição de botnet no Capítulo 11.

<sup>2</sup>gcc: *GNU Compiler Collection*.

Dev-C++ deve ser empregado. O Dev-C++ possui o gcc embutido, por padrão. Não se faz necessário a instalação de pacotes adicionais de modo a se criar aplicativos utilitários em ANSI-C. Apesar de descontinuado, o Dev-C++ continua sendo largamente empregado como ambiente de desenvolvimento em ANSI-C. Alternativamente, podem ser utilizados outros ambientes tal qual o *Code::Blocks* e *Microsoft Visual Studio*.

## 12.2 Aplicativo em segundo plano

Grande parte dos *malware* não visam ser detectados pela vítima. Por exemplo, mineradores de bitcoins e cavalos de troia não desejam ser percebidos pela vítima. Ao contrário estão os *ransomware* que são visualmente detectáveis conforme visto no Cap. 1. Um executável pode estar em segundo plano. Logo o console (tela preta) não irá aparecer para o usuário. A biblioteca "Windows.h" necessária para carregar a função "FreeConsole()", empregada para esconder o console.

### Siga as instruções:

- 1 Faça o *Download* da pasta *esconder\_console*, responsável por criar e escrever texto em arquivo, no [presente link](#) (pasta Cap. 13).
- 2 Faça a abertura do *script* *esconder\_console.cpp* pelo Dev C++. **Pressione F9** para compilar o código-fonte *esconder\_console.cpp*. Na sequência, execute o aplicativo dando duplo-clique sobre ele, como mostra a Fig. 12.1.

```
9  #include <stdio.h>
10 #include <stdlib.h>
11 #include <Windows.h>
12
13 int main() {
14     FreeConsole();
15     while(true){
16     }
17     return 0;
18 }
```

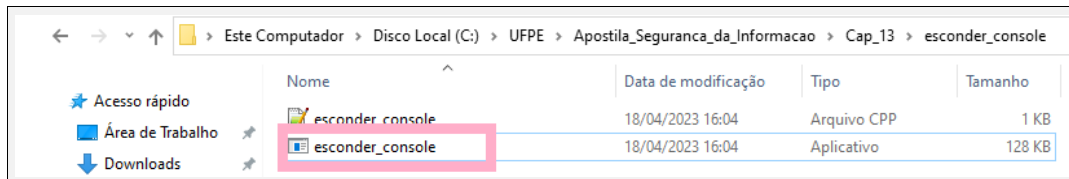
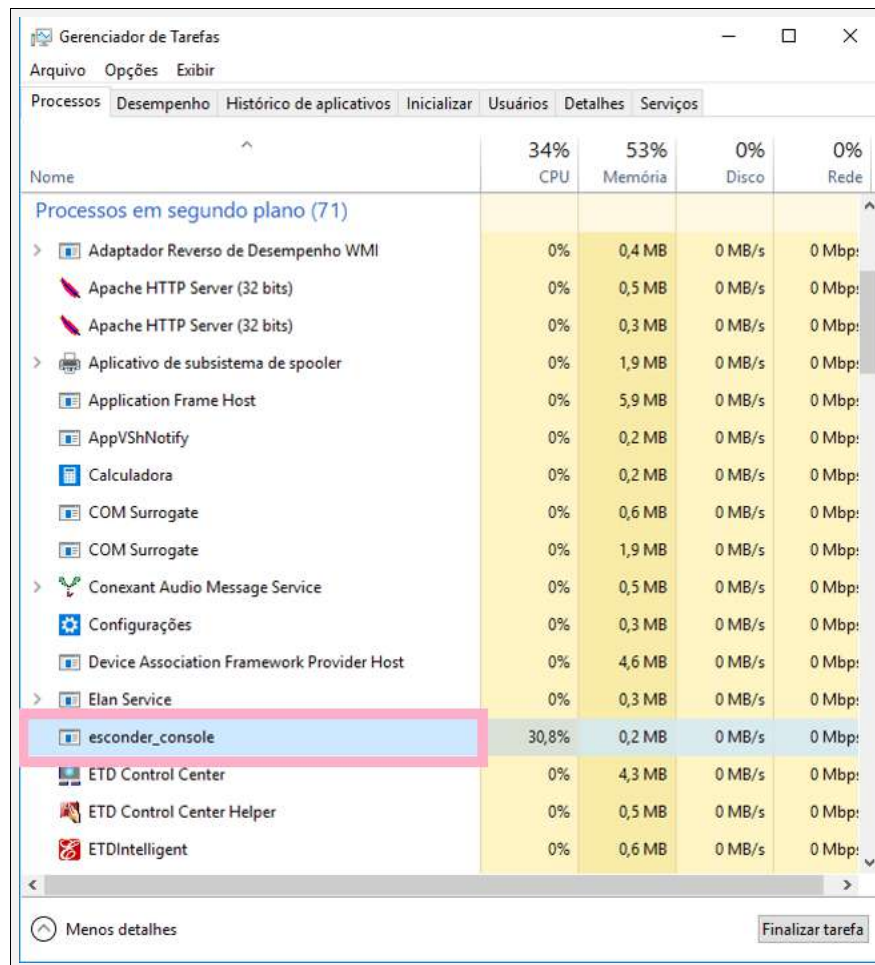


Figura 12.1: Dê um duplo-clique para a execução do aplicativo.

3 A Fig. 12.2 apresenta o executável funcionando em segundo plano.



Nome	34% CPU	53% Memória	0% Disco	0% Rede
<b>Processos em segundo plano (71)</b>				
Adaptador Reverso de Desempenho WMI	0%	0,4 MB	0 MB/s	0 Mbps
Apache HTTP Server (32 bits)	0%	0,5 MB	0 MB/s	0 Mbps
Apache HTTP Server (32 bits)	0%	0,3 MB	0 MB/s	0 Mbps
Aplicativo de subsistema de spooler	0%	1,9 MB	0 MB/s	0 Mbps
Application Frame Host	0%	5,9 MB	0 MB/s	0 Mbps
AppVShNotify	0%	0,2 MB	0 MB/s	0 Mbps
Calculadora	0%	0,2 MB	0 MB/s	0 Mbps
COM Surrogate	0%	0,6 MB	0 MB/s	0 Mbps
COM Surrogate	0%	1,9 MB	0 MB/s	0 Mbps
Conexant Audio Message Service	0%	0,5 MB	0 MB/s	0 Mbps
Configurações	0%	0,3 MB	0 MB/s	0 Mbps
Device Association Framework Provider Host	0%	4,6 MB	0 MB/s	0 Mbps
Elan Service	0%	0,3 MB	0 MB/s	0 Mbps
<b>esconder_console</b>	<b>30,8%</b>	<b>0,2 MB</b>	<b>0 MB/s</b>	<b>0 Mbps</b>
ETD Control Center	0%	4,3 MB	0 MB/s	0 Mbps
ETD Control Center Helper	0%	0,5 MB	0 MB/s	0 Mbps
ETDIntelligent	0%	0,6 MB	0 MB/s	0 Mbps

Figura 12.2: Aplicativo sendo executado em segundo plano.

### 12.3 Criar executável com ícone

Uma característica comum dos *malware* é tentar se fazer passar por um aplicativo sério. Nesse contexto, usar um ícone de um aplicativo sério pode induzir a vítima a não atentar que seu computador está infectado.

#### Siga as instruções:

- 1 Faça o *Download* da pasta *executavel\_com\_icone*, responsável por criar um aplicativo com ícone, no [presente link](#) (pasta Cap. 13).
- 2 Configuração do ambiente de desenvolvimento Dev C++.
  - Fig.12.3: clique no menu "**Arquivo**", na sequência clique em "**Novo**" → "**Projeto...**".
  - Fig.12.4: escolha a opção "**Console Application**" e clique em "**Ok**". *Console Application* significa a criação de um aplicativo com o console (sem interface gráfica).
  - Fig.12.5: clique em "**Salvar**".
  - Fig.12.6: **pressione F11** para compilar e executar o aplicativo. O objetivo é verificar se o projeto foi criado corretamente. Na sequência, feche o *console*.
  - Fig.12.7: clique no menu "**Projeto**", na sequência clique em "**Opções do Projeto**". O objetivo é incorporar a biblioteca gráfica ao projeto.
  - Fig.12.8: Na guia "**General**", escolha "**Browse**". Por fim, escolha o \*.ico.
  - **Pressione F9** para compilar o aplicativo final. Na sequência, execute o aplicativo dando duplo-clique sobre ele.

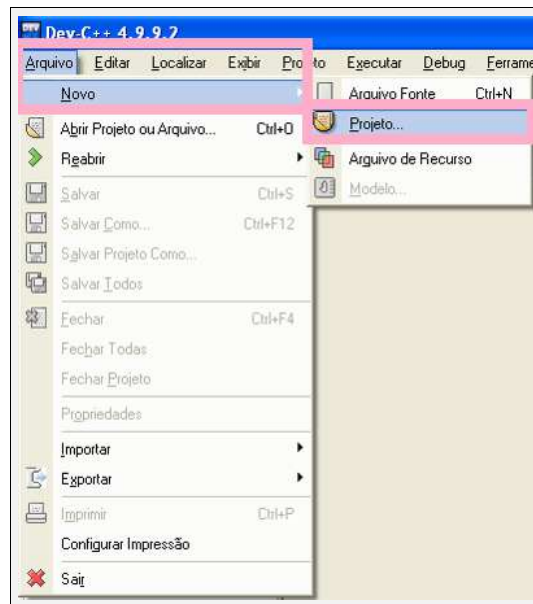


Figura 12.3: Clique no menu "**Arquivo**", na sequência clique em "**Novo**" → "**Projeto...**".

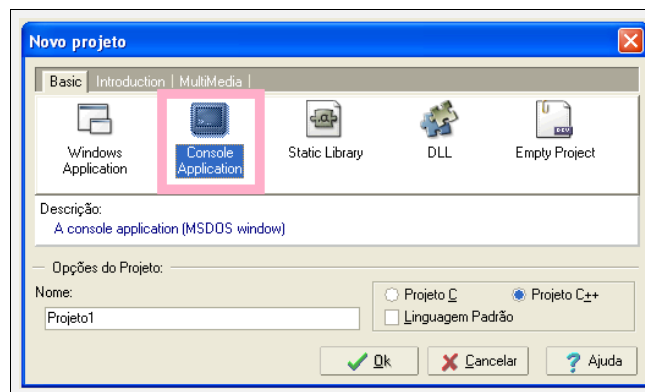


Figura 12.4: Escolha a opção *"Console Application"* e clique em *"Ok"*.

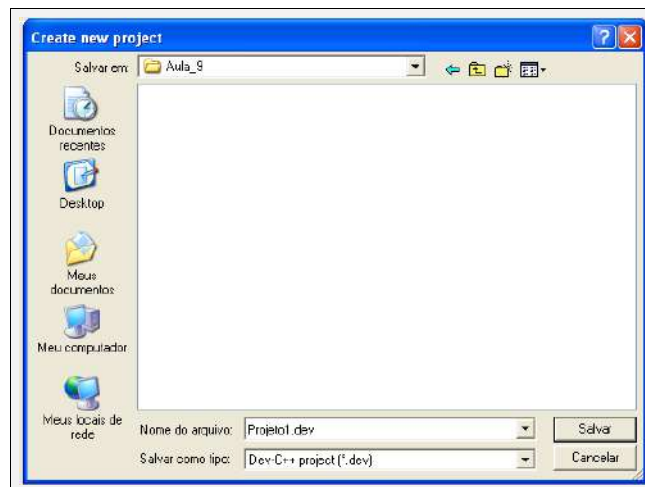


Figura 12.5: Clique em *"Salvar"*.

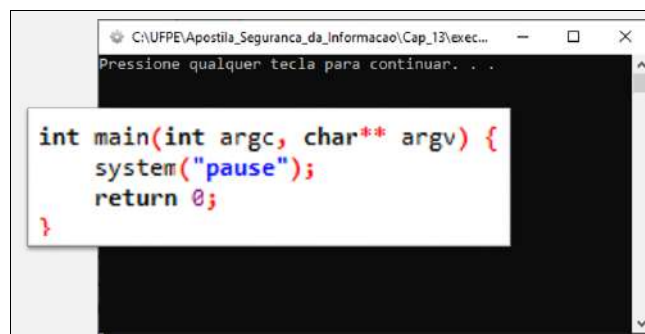


Figura 12.6: **Pressione F11** para compilar e executar o aplicativo. O objetivo é verificar se o projeto foi criado corretamente. Na sequência, feche o *console*.



Figura 12.7: Clique no menu **"Projeto"**, na sequência clique em **"Opções do Projeto"**.

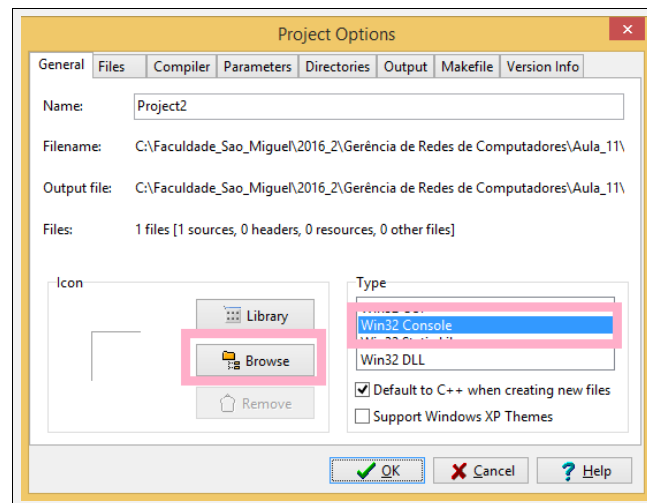


Figura 12.8: Na guia **"General"**, escolha **"Browse"**. Por fim, escolha o \*.ico.

## 12.4 Criar executável dotado de *Multimídia*

A função *mciSendString*<sup>3</sup> envia uma *string* para um dispositivo MCI (*Media Control Interface* – Interface para Controle de Mídia). Tecnicamente, um dispositivo mci é uma API desenvolvido pelo Microsoft e IBM para o controle de periféricos multimídia como:

- controlador de áudio,
- *drive* de CD-ROM.

**Siga as instruções:**

- 1 Faça o *Download* da pasta *multimidia*, responsável por criar um aplicativo dotado de música, no presente link (pasta Cap. 13).
- 2 Configuração do ambiente de desenvolvimento Dev C++.
  - Fig.12.9: clique no menu "**Arquivo**", na sequência clique em "**Novo**" → "**Projeto...**".
  - Fig.12.10: escolha a opção "**Console Application**" e clique em "**Ok**". *Console Application* significa a criação de um aplicativo com o console (sem interface gráfica).
  - Fig.12.11: clique em "**Salvar**".
  - Fig.12.12: **pressione F11** para compilar e executar o aplicativo. O objetivo é verificar se o projeto foi criado corretamente. Na sequência, feche o *console*.
  - Fig.12.13: clique no menu "**Projeto**", na sequência clique em "**Opções do Projeto**". O objetivo é incorporar a biblioteca gráfica ao projeto.
  - Fig.12.14: na guia "**General**", escolha "**Browse**". Por fim, escolha o \*.ico. No campo "**Linker**", acrescente o termo *-lwinmm*.

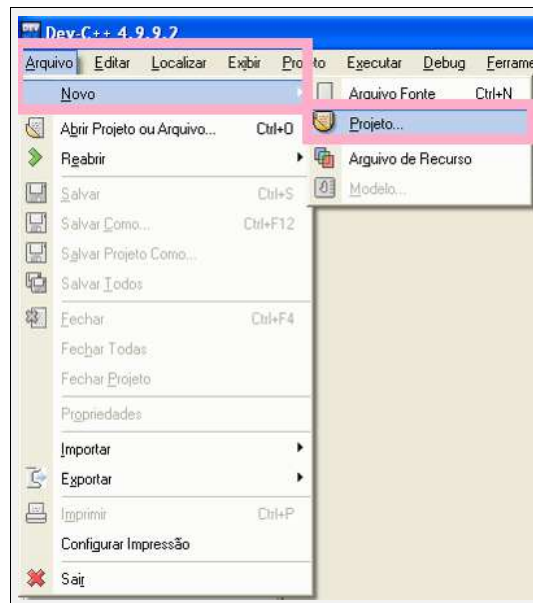


Figura 12.9: Clique no menu "**Arquivo**", na sequência clique em "**Novo**" → "**Projeto...**".

<sup>3</sup>Função *mciSendString*. Disponível em: [https://msdn.microsoft.com/pt-br/library/windows/desktop/dd757161\(v=vs.85\).aspx](https://msdn.microsoft.com/pt-br/library/windows/desktop/dd757161(v=vs.85).aspx). Acesso em março de 2022.



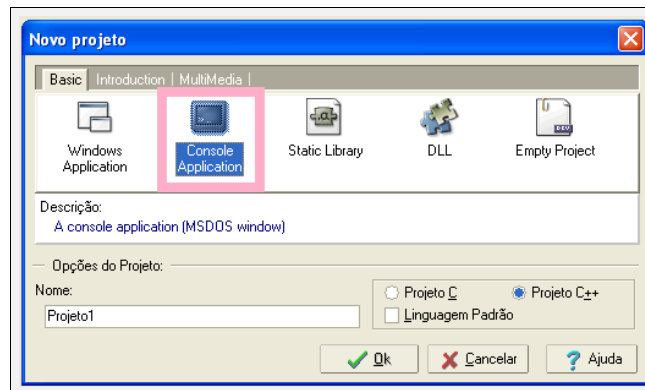


Figura 12.10: Escolha a opção *"Console Application"* e clique em *"Ok"*.

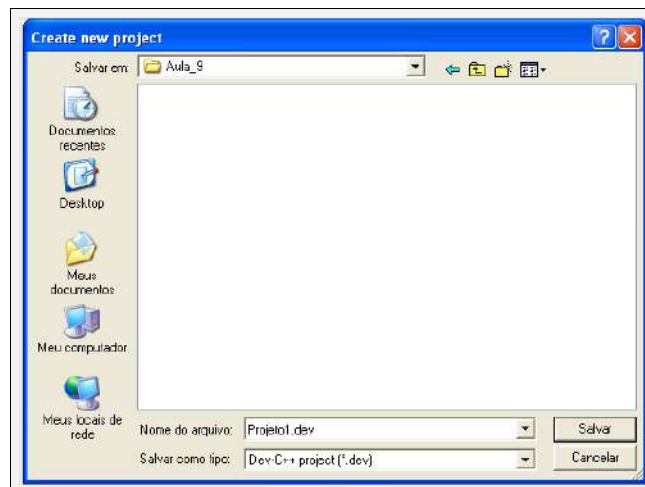


Figura 12.11: Clique em *"Salvar"*.

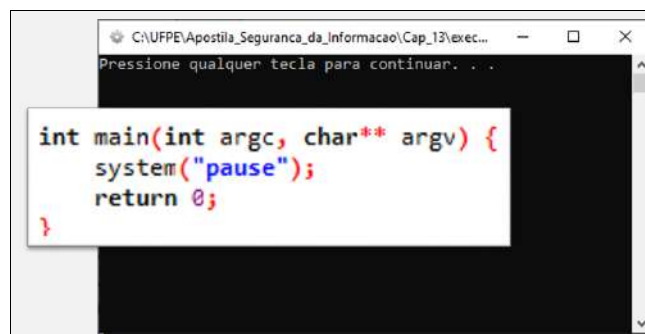


Figura 12.12: **Pressione F11** para compilar e executar o aplicativo. O objetivo é verificar se o projeto foi criado corretamente. Na sequência, feche o *console*.



Figura 12.13: Clique no menu "**Projeto**", na sequência clique em "**Opções do Projeto**".

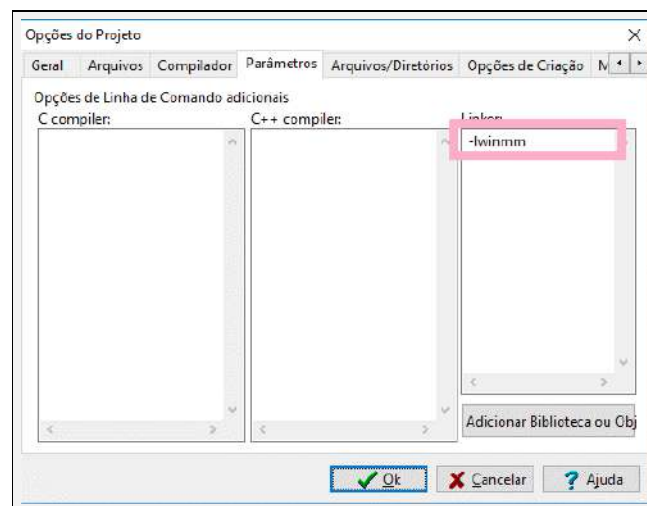


Figura 12.14: Na guia "**General**", escolha "**Browse**". Por fim, escolha o \*.ico. No campo "**Linker**", acrescente o termo *-lwinmm*.

**3** Multimídia: executar uma música automaticamente.

- O arquivo de música \*.mp3 deve estar na mesma pasta do executável. Na linha 18, primeiro o áudio é aberto. Na linha 20, o arquivo \*.mp3 é executado de forma repetida através “repeat”. Caso não houvesse o comando “repeat”, a música seria executada uma única vez.

```
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <windows.h> // para PlaySound()
11
12 int main(int argc, char** argv) {
13
14     // -----Esconder a tela do console no
15     //-----no sistema operacional Windows -----
16     FreeConsole();
17
18     //-----abre o áudio-----
19     mciSendString("open badboys.mp3 type mpegvideo", NULL, 0,0);
20     //-----A música de fundo é tocada repetitivamente-----
21     mciSendString("play badboys.mp3 repeat" , NULL, 0, 0);
22
23     while(true){
24     }
25
26     return 0;
27 }
```

- **Pressione F9** para compilar o aplicativo final. Na sequência, execute o aplicativo dando duplo-clique sobre ele.

## 12.5 Cavalo de Troia em dispositivo de CD-ROM

A função *mciSendString*<sup>4</sup> envia uma *string* para um dispositivo MCI (*Media Control Interface* – Interface para Controle de Mídia). Tecnicamente, um dispositivo mci é uma API desenvolvido pelo Microsoft e IBM para o controle de periféricos multimídia como:

- controlador de áudio,
- *drive* de CD-ROM.

**Siga as instruções:**

- 1 Faça o *Download* da pasta *trojan\_cd\_rom*, responsável por criar um aplicativo capaz de manipular o dispositivo (*driver*) de CD-ROM, no [presente link](#) (pasta Cap. 13).
- 2 Configuração do ambiente de desenvolvimento Dev C++.
  - Fig.12.15: clique no menu "**Arquivo**", na sequência clique em "**Novo**" → "**Projeto...**".
  - Fig.12.16: escolha a opção "**Console Application**" e clique em "**Ok**". *Console Application* significa a criação de um aplicativo com o console (sem interface gráfica).
  - Fig.12.17: clique em "**Salvar**".
  - Fig.12.18: **pressione F11** para compilar e executar o aplicativo. O objetivo é verificar se o projeto foi criado corretamente. Na sequência, feche o *console*.
  - Fig.12.19: clique no menu "**Projeto**", na sequência clique em "**Opções do Projeto**". O objetivo é incorporar a biblioteca gráfica ao projeto.
  - Fig.12.20: na guia "**General**", escolha "**Browse**". Por fim, escolha o \*.ico. No campo "**Linker**", acrescente o termo *-lwinmm*.

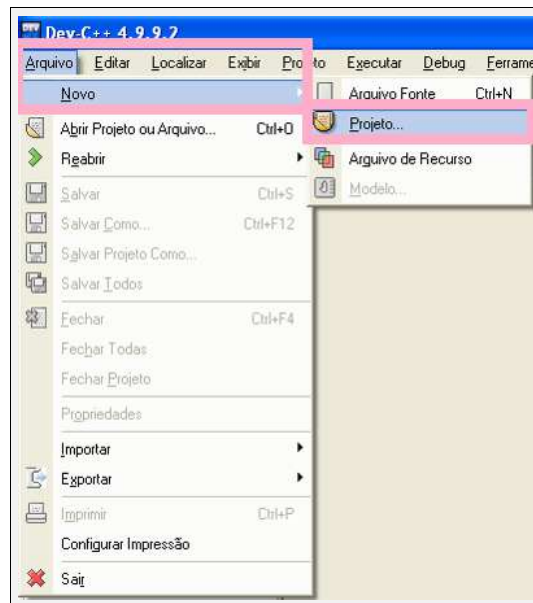


Figura 12.15: Clique no menu "**Arquivo**", na sequência clique em "**Novo**" → "**Projeto...**".

<sup>4</sup>Função *mciSendString*. Disponível em: [https://msdn.microsoft.com/pt-br/library/windows/desktop/dd757161\(v=vs.85\).aspx](https://msdn.microsoft.com/pt-br/library/windows/desktop/dd757161(v=vs.85).aspx). Acesso em março de 2022.

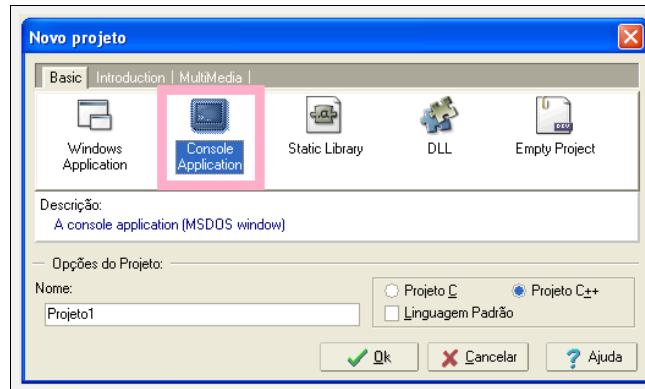


Figura 12.16: Escolha a opção *"Console Application"* e clique em *"Ok"*.

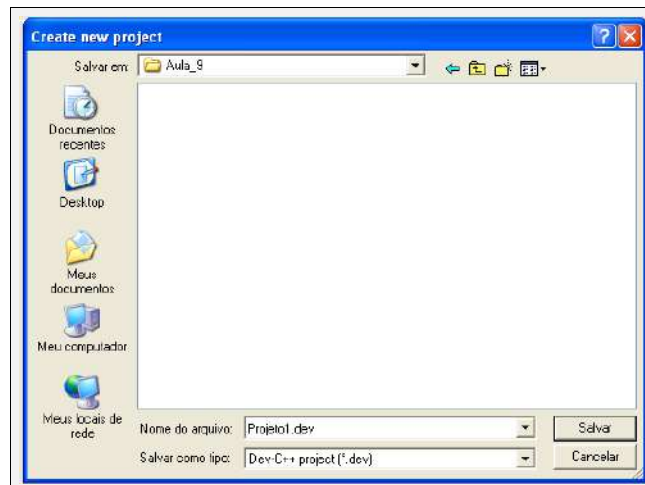


Figura 12.17: Clique em *"Salvar"*.

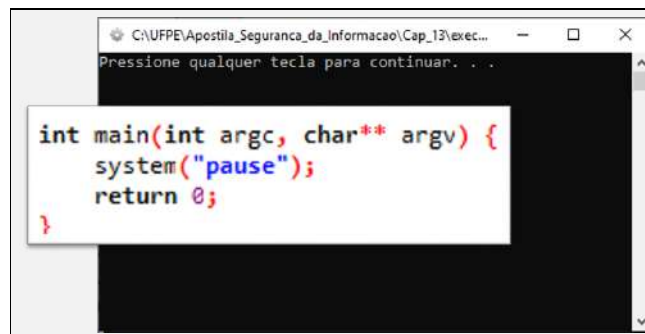


Figura 12.18: **Pressione F11** para compilar e executar o aplicativo. O objetivo é verificar se o projeto foi criado corretamente.



Figura 12.19: Clique no menu "**Projeto**", na sequência clique em "**Opções do Projeto**".

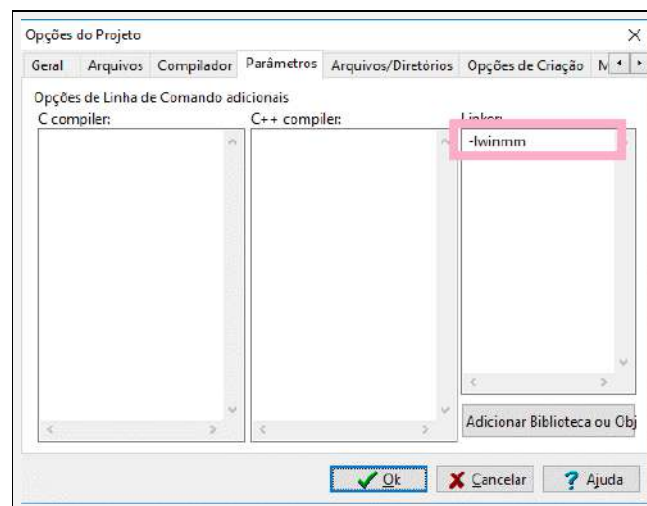


Figura 12.20: Na guia "**General**", escolha "**Browse**". Por fim, escolha o \*.ico. No campo "**Linker**", acrescente o termo *-lwinmm*.

3 Dentro da pasta *trojan\_cd\_rom*, há o script *main.cpp*.

- **"open cdaudio"**: aloca o dispositivo de CD-ROM ao executável.
- **"set cdaudio door open"**: abre o drive de CD-ROM.
- **"set cdaudio door closed"**: fecha o drive de CD-ROM.
- **"close cdaudio"**: desaloca o dispositivo de CD-ROM ao executável. Desse modo, outros executáveis poderão alocar o drive de CD-ROM.
- A função **"Sleep"**<sup>5</sup> suspende o executável até o tempo passado como parâmetro em mili-segundos.

```

8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <Windows.h>    // para PlaySound()
11
12 int main(int argc, char** argv) {
13     // -----Esconder a tela do console no
14     //-----no sistema operacional Windows -----
15     FreeConsole();
16
17     while(true){
18         //-----abre o cd rom-----
19         mciSendString("open cdaudio", 0, 0, 0);
20         mciSendString("set cdaudio door open", 0, 0, 0);
21         //-----O programa fica preso (dormindo)---
22         //-----por 2 segundos-----
23         Sleep(2000);
24         //-----fecha o cd rom-----
25         mciSendString("set cdaudio door closed", 0, 0, 0);
26         mciSendString("close cdaudio", 0, 0, 0);
27         //-----O programa fica preso (dormindo)---
28         //-----por 2 segundos-----
29         Sleep(2000);
30     }
31     return 0;
32 }

```

- **Pressione F9** para compilar o aplicativo final. Na sequência, execute o aplicativo dando duplo-clique sobre ele.

<sup>5</sup>Função *Sleep*. Disponível em: [https://msdn.microsoft.com/pt-br/library/windows/desktop/ms686298\(v=vs.85\).aspx](https://msdn.microsoft.com/pt-br/library/windows/desktop/ms686298(v=vs.85).aspx). Acesso em março de 2023.

## 12.6 Multimídia + Esteganografia

A esteganografia pode ser empregada em conjunto com um arquivo multimídia. Isso quer dizer, um aplicativo invocar um arquivo escondido em formato de música/vídeo. Trata-se de um aplicativo de advertência, sem efeitos nocivos práticos. O objetivo é demonstrar as vulnerabilidades expostas e que poderiam causar malfeitorias, possivelmente, irreversíveis e irrecuperáveis.

① É necessário estudar, de maneira prévia, a seção prévia sobre esteganografia (*Alternate Data Stream*). Na pasta "*primeira\_parte\_multimidia*", é criado um executável visando tocar o arquivo .mp3 oculto (escondido) no executável. Abra o projeto "*malicioso\_de\_advertencia.dev*" e **pressione F9** apenas para Compilar. Não execute pois ainda não há o arquivo .mp3 oculto acoplado ao executável.

② Copie o executável "*malicioso\_de\_advertencia.exe*" para a pasta *segunda\_parte\_multimidia*. Abra o *esteganografia.dev* e **pressione F11** para compilar e executar. Logo o arquivo .mp3 será oculto (escondido) no executável no "*malicioso\_de\_advertencia.exe*".

③ O "*malicioso\_de\_advertencia.exe*" está pronto para ser usado, copie para um lugar diverso e o execute. Por exemplo, o executável pode ser copiado para a pasta "*terceira\_parte\_executavel\_pronto*". No *prompt* de comando, é possível verificar a existência de um arquivo oculto.

- **Pressione *shift***, com o lado direito do mouse escolha a opção "**Abrir janela do PowerShell aqui**". Digite o seguinte comando:

```
dir /r
```

④ Um determinado aplicativo dotado de um arquivo multimídia (música) escondido não é mais transferível para um pen-drive minimamente atual. Em síntese, a combinação entre multimídia e esteganografia é inviável nos pen-drives atuais. Restaria a validade do experimento em pen-drives antigos. Faça o seguinte experimento:

- Copie o aplicativo "*malicioso\_de\_advertencia.exe*" para o pen-drive.
- Na partição do pen-drive, **pressione *shift***, com o lado direito do mouse escolha a opção "**Abrir janela do PowerShell aqui**". Digite o seguinte comando:

```
dir /r
```

- No *prompt* de comando, é possível notar que o o arquivo oculto foi eliminado automaticamente pelo *firmware* do próprio pen-drive.
  - Por *firmware*, denota-se um *software* capaz de comandar o *hardware* sem a concessão do sistema operacional. Convencionalmente, o *firmware* é armazenamento e processado no próprio periférico quando acionado.