UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

INSTITUTO METROPOLE DIGITAL

Disciplina: IMD0025 - Laboratório de Programação I

Aula 3 – Depuração de Código

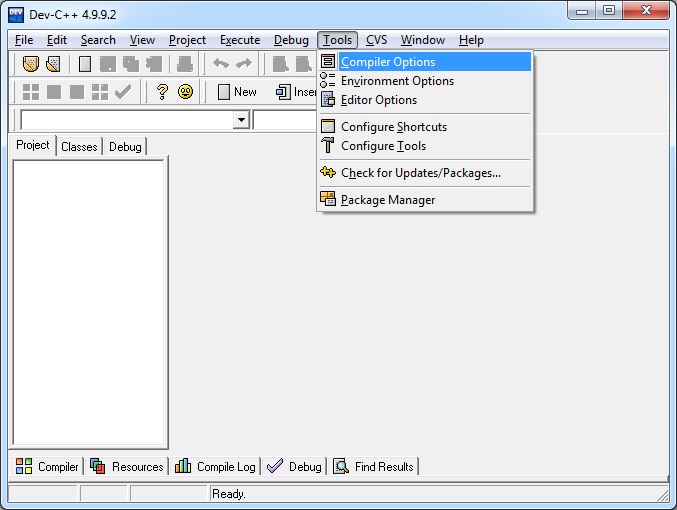
**Tutorial: Depurando código com GDB via Dev-C++ ou via Terminal de Comandos**

**1. Ambiente Dev-C++ (versão 4.9.9.2 – 32 bits)**

Para fazermos a depuração de código escrito em linguagem C++ utilizando o ambiente Dev-C++ 4.9.9.2, faz-se necessário configurar alguns parâmetros para que o ambiente de depuração funcione corretamente.

**1.1. Configuração do compilador G++ no Dev-C++ 4.9.9.2**

Para configurar o ambiente Dev-C++ para depuração, vá ao menu ***Tools*** (Ferramentas) | ***Compiler Options*** (Opções do Compilador), conforme a *Figura 1*.



*Figura 1: Menu principal*

Ao clicar na opção ***Compiler Options***, uma nova tela será aberta, conforme a *Figura 2*. Na aba ***Compiler*** desta tela, selecione as duas caixas de seleção com os nomes ***Add the following commands when calling compiler*** e ***Add these commands to the linker command line***, e adicione em ambos os campos a seguinte diretiva de compilação e de linkedição:

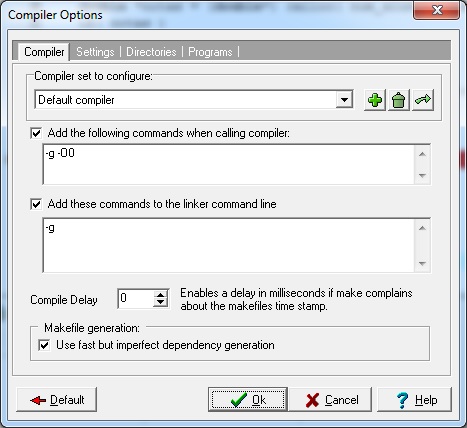
-g (sinal de subtração, letra “g” minúscula)

Isto permitirá indicar ao compilador e ao linkeditor que queremos inserir as informações de depuração no código de máquina a ser gerado.

Em seguida, insira apenas no campo da caixa de seleção ***Add the following commands when calling compiler***, a seguinte diretiva de compilação precedida de um espaço:

-O0 (sinal de subtração, letra “O” maiúscula, zero)

Isto permitirá indicar ao compilador que toda forma de otimização, que poderia impedir o depurador de funcionar corretamente, deve ser eliminada.



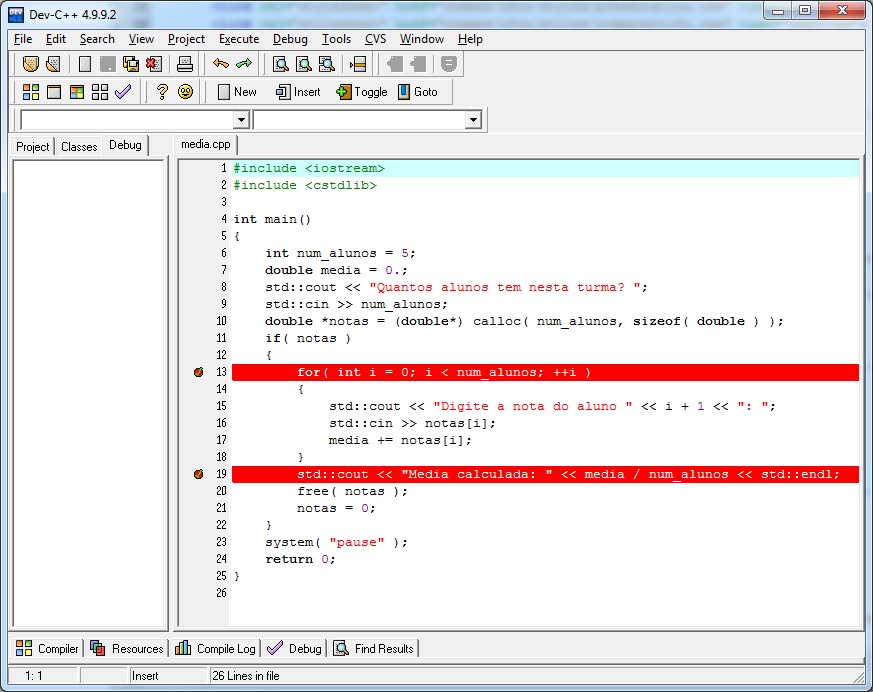
*Figura 2: Configuração das diretivas de compilação nas opções do compilador*

Para confirmar as alterações, basta clicar no botão ***Ok***. A configuração do compilador e do linkeditor estarão prontas para a depuração.

**1.2. Execução do depurador de código no Dev-C++ 4.9.9.2**

Para depurar o código de um programa no Dev-C++, é necessário abrir neste ambiente o arquivo do referido código e marcar os devidos *breakpoints* (pontos de parada) que permitirão fazermos em seguida um passo a passo.

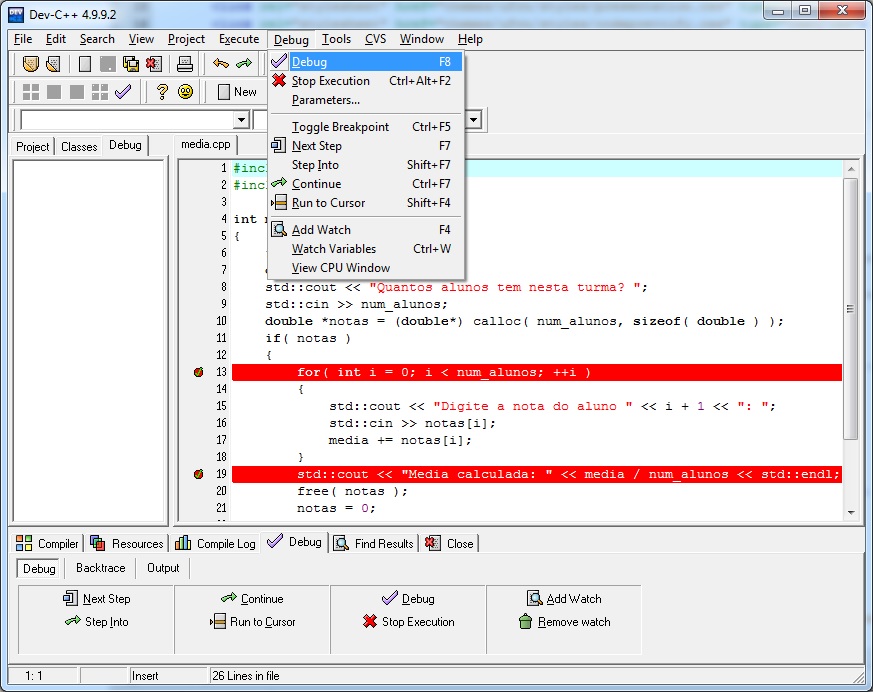
Para tanto, basta clicar na coluna que se situa à esquerda do código fonte no ambiente do Dev-C++, na altura da linha onde se pretende interromper a execução do código. Imediatamente aparecerá uma faixa vermelha sobre a linha escolhida, evidenciando o ponto de parada. Aparecerá também uma bolinha vermelha na coluna onde se clicou, conforme a *Figura 3*.



*Figura 3: Exemplos de pontos de parada*

Para executar o código em modo de depuração, basta pressionar a tecla **F8** ou escolher a opção ***Debug*** no menu ***Debug***. Conforme a *Figura 4*, podemos notar que o menu ***Debug*** possui também outras opções fundamentais para a depuração do código. Estas opções são as seguintes:

* **Debug** (F8): executar o código em modo depuração;
* **Stop Execution** (Ctrl + Alt +F2): parar a depuração;
* **Parameters...**: adicionar parâmetros de depuração e otimização;
* **Toggle Breakpoint** (Ctrl + F5): marcar pontos de parada no código;
* **Next Step** (F7): executar o código linha por linha;
* **Step Into** (Shift + F7): entrar no código das funções quando for possível;
* **Continue** (Ctrl + F7): executar o código até encontrar um ponto de parada;
* **Run to Cursor** (Shift + F4): executar o código até encontrar o cursor;
* **Add Watch** (F4): visualizar os valores dos objetos e variáveis do programa. Para isto, apertar a tecla F4 e digitar, na caixa de diálogo que aparecerá, os nomes das variáveis que deverão ser observadas. Ao adicionar uma variável para observação, ela será mostrada na aba *Debug* localizada no lado esquerdo da interface do Dev-C++;
* **Watch Variables** (Ctrl + W): visualizar a aba *Debug* caso esta ainda não esteja visível;
* **View CPU Window**: mostrar o que está acontecendo no processador do computador, o que consiste nos comandos que serão executados em linguagem Assembly e o que está armazenado nos registradores do processador no momento da execução.



*Figura 4: Opções do menu Debug*

Uma vez que sabemos para que serve cada opção do menu ***Debug***, podemos começar a depurar nossos programas.

Para começar a depuração, é necessário que o código a ser depurado esteja compilado, o que pode ser feito através das teclas **Ctrl + F9** ou via a opção de compilação no menu ***Execute*** | ***Compile***. Em seguida, se os pontos de parada já tiverem sido marcados no código fonte, pressione a tecla **F8** ou clique no menu ***Debug*** | ***Debug*** para execução do código em modo depuração.

O programa será executado normalmente até que seja encontrado um ponto de parada ou um comando de leitura de dados (por exemplo, std::cin). Caso um comando de leitura de dados seja executado, a tela de entrada de dados será inicializada e o compilador aguardará até que o valor seja digitado e a tecla ***Enter*** seja pressionada. Feito isto, volte ao ambiente de depuração e continue a depuração do código.

Para executar o código linha por linha, pressione a tecla **F7** ou clique no menu ***Debug*** | ***Next Step***. Para entrar no código de uma função, deve-se continuar a depuração até chegar na linha da função em questão e pressionar as teclas **Shift + F7** ou clicar na opção ***Step Into*** no menu ***Debug***. Dentro da função, o código pode ser depurado linha por linha. Caso queira observar o valor de alguma variável, adicione-a à tela de observação pressionando a tecla **F4** ou clicando com o botão direito do *mouse* em cima da variável a fim de escolher a opção ***Add Watch***.

**2. Ambiente Dev-C++ (versão 5.4.2 – 32 e 64 bits)**

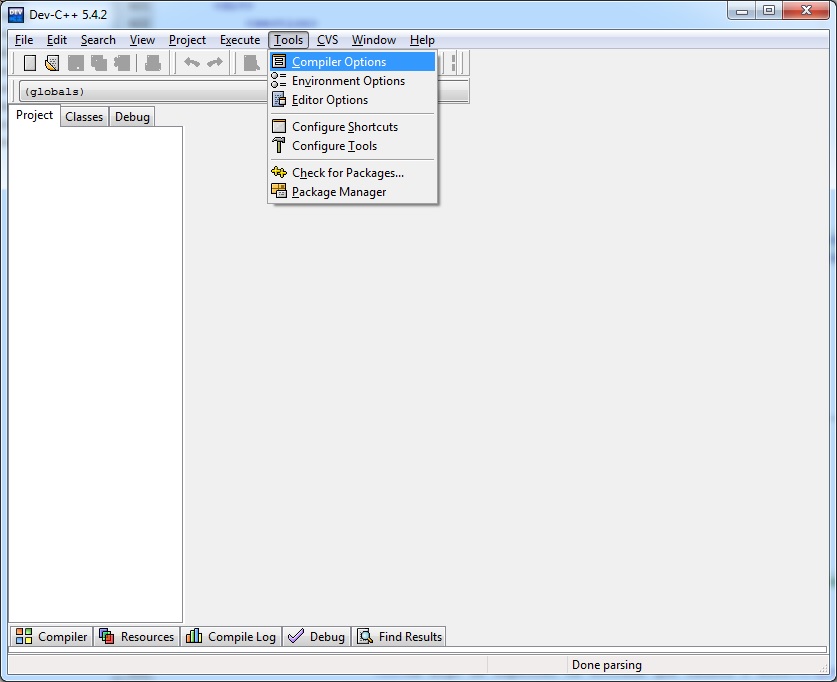
Como na versão antiga do Dev-C++, para fazermos a depuração de código escrito em linguagem C++ utilizando a versão mais recente do ambiente Dev-C++ (versão 5.4.2), faz-se necessário configurar alguns parâmetros para que o ambiente de depuração funcione corretamente.

**2.1. Configuração do compilador G++ no Dev-C++ 5.4.2**

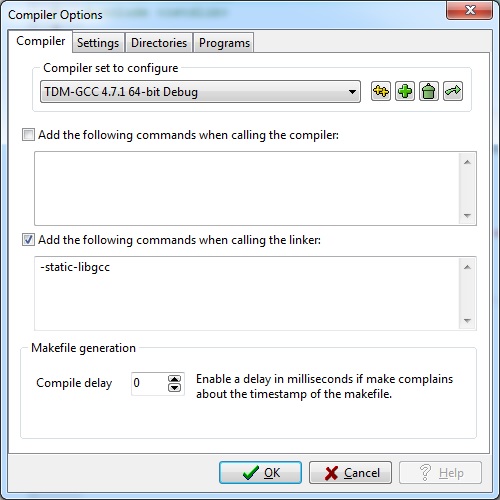
Para configurar o ambiente Dev-C++ para depuração, vá ao menu ***Tools*** (Ferramentas) | ***Compiler Options*** (Opções do Compilador), conforme a *Figura 5*.

Ao clicar na opção ***Compiler Options***, uma nova tela será aberta, conforme a *Figura 6*. Na aba ***Compiler*** desta tela, selecionar a opção ***64-bit Debug*** ou ***32-bit Debug*** da lista ***Compiler set to configure***. Esta seleção, que é bem mais simples que na versão antiga do Dev-C++, será suficiente para configurar o compilador e o linkeditor para o procedimento de depuração.

Para confirmar as alterações, basta clicar no botão ***Ok***. A configuração do compilador e do linkeditor estarão prontas para a depuração.



*Figura 5: Menu principal*



*Figura 6: Configuração das diretivas de compilação nas opções do compilador*

**2.2. Execução do depurador de código no Dev-C++ 5.4.2**

Para depurar o código de um programa no Dev-C++ 5.4.2, é necessário abrir neste ambiente o arquivo do referido código e marcar os devidos *breakpoints* (pontos de parada) que permitirão fazermos em seguida um passo a passo.

Para executar o código em modo de depuração, basta pressionar a tecla **F5** ou escolher a opção ***Debug*** no menu ***Execute***. Diversos butões para manipulação são ativados na parte de baixo do ambiente. A lista de opções disponíveis se parece bastante com aquela apresentada na seção 1.2 deste tutorial.

**3. Terminal de comandos**

Para fazermos a depuração de código escrito em linguagem C++ utilizando o terminal de comandos, faz-se necessário introduzir os comandos do depurador **gdb** e sua sintaxe e as diretivas do compilador e do linkeditor.

**3.1. Configuração do compilador g++ no terminal de comandos**

Tal qual no Dev-C++, utilizaremos as diretivas de compilação e de linkedição que permitirão indicar ao compilador e ao linkeditor que queremos inserir as informações de depuração no código de máquina a ser gerado, como também indicar ao compilador que toda forma de otimização, que poderia impedir o depurador de funcionar corretamente, deve ser eliminada.

Para isto, ao compilarmos um programa, utilizaremos a seguinte sintaxe:

g++ -g -O0 programa.cpp -o programa

A diretiva **-g** permite inserir as informações de depuração e a diretiva **-O0** permite eliminar a otimização.

**3.2. Execução do depurador de código no terminal de comandos**

Para executar o código de um programa em modo de depuração através do terminal de comandos, é necessário que o código esteja compilado. Faz-se portanto uma chamada ao depurador **gdb** usando como parâmetro o nome do programa compilado:

gdb programa

Uma vez a execução do depurador iniciada, podemos utilizar diversas opções fundamentais para a depuração do código. Estas opções são as seguintes:

* **q**: sai do **gdb**;
* **r**: executa o programa do início;
* **l**: lista porções de código;
* **b** *numero*: insere um ponto de parada na linha *numero*;
* **d** *numero*: apaga o ponto de parada da linha *numero*;
* **n**: executa a próxima instrução;
* **p** *variavel*: imprime o valor atual da variável *variavel*;
* **c**: continua a execução do programa sem paradas.

Para mais informações, verificar o guia de referência rápida para o **GDB,** que pode ser encontrado no SIGAA, e acessar os seguintes tutoriais na internet:

* <http://www.yolinux.com/TUTORIALS/GDB-Commands.html>
* <http://jumpi.wordpress.com/2007/07/24/tutorial-basico-de-gdb/>
* <http://www.dei.isep.ipp.pt/~jcoelho/doc/gdb.html>