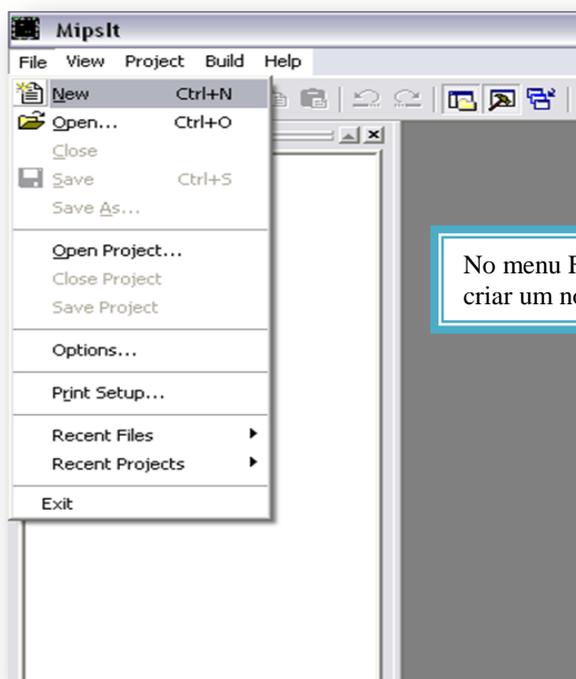


Usando o simulador MIPS

O objetivo desta aula prática será a utilização do simulador MipsIt para executar programas escritos em linguagem de máquina do MIPS.

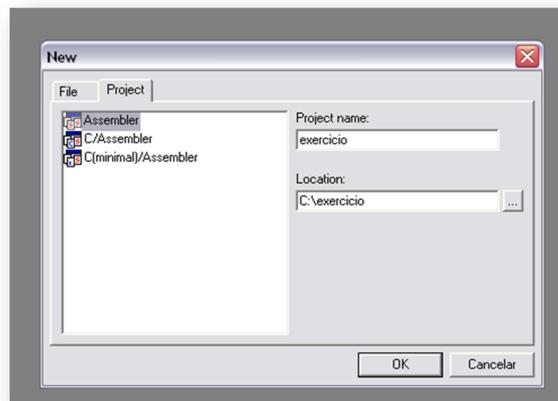
1 – Criando um projeto

Cada programa a ser executado deverá estar inserido dentro do contexto de um projeto. Para criar um projeto execute o programa MipsIt.exe.



No menu File clique em New para criar um novo projeto.

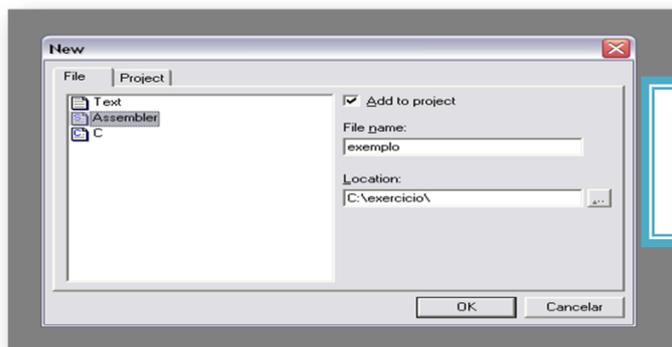
Neste momento você deve selecionar o diretório de localização do projeto usando o botão ao lado de Location. Lembre-se de especificar o nome que o projeto deve receber dentro da caixa Project name.



Agora seu projeto acaba de ser criado. Devemos passar para a próxima etapa.

2 – Editando um programa em linguagem de máquina:

Agora devemos criar um arquivo assembly para que possamos observar o funcionamento do programa. Para tanto vá ao menu File e selecione a opção New. Neste momento a seguinte caixa de texto surgirá na tela.



Escolha a opção Assembler entre as três apresentadas. Na opção File name indique o nome que seu arquivo deve receber (neste tutorial ele irá se chamar exemplo).

Use o programa abaixo para seguir neste tutorial. Copie-o e salve o arquivo e o projeto.

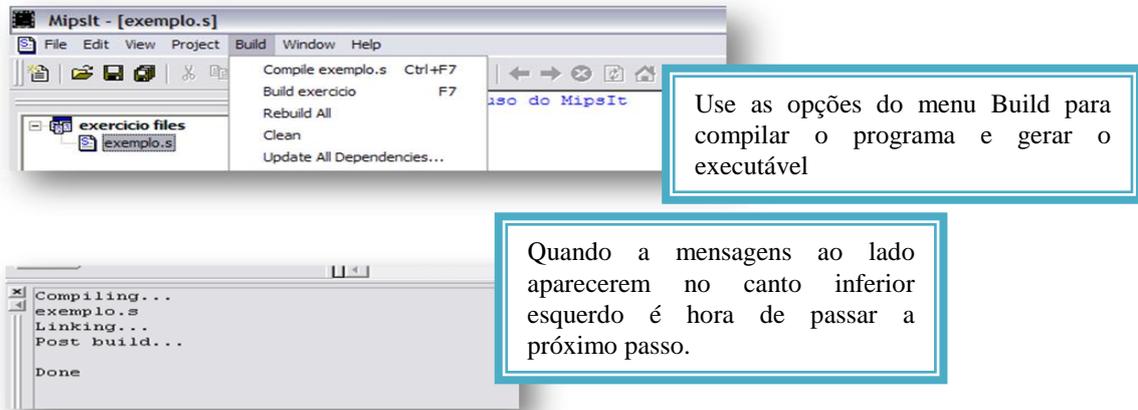
```
.data
a:    .word 5
b:    .word 10
c:    .word 5
      .text
      .globl start
      .ent start
start: lw    $8, a
      lw    $9, b
      lw    $10, c
      add   $11, $9, $8
      sub   $11, $11, $10
      sw    $11, a
      .end start
```

Os construtores iniciando com ponto (.) são pseudo instruções ou diretivas que tem como função dar algumas indicações ao montador sem que gere código executável.

O construtor de montagem *.data* reserva área de memória para as variáveis *a*, *b* e *c*, que são do tipo inteiro de 32 bits (*.word*). O construtor *.text* delimita início de código e os construtores *.globl start* e *.ent start* declaram *start* como endereço simbólico de início de programa. A partir do endereço simbólico *start* tem-se um programa em linguagem de máquina. O construtor *.end start* delimita o fim do programa cujo endereço inicial simbólico foi declarado como *start*. Nossa próxima etapa é compilar o programa acima.

3- Compilando um programa em *assembly*.

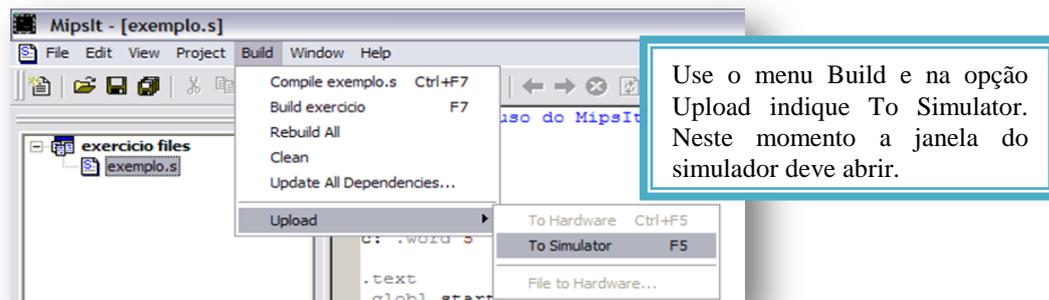
Para compilar o programa escolha a opção *compile* no menu *build*. Após corrigir os erros de sintaxe, o código executável poderá ser gerado através da opção *build* no menu *build*. Os códigos objeto e executável serão armazenados no diretório de projeto.



Atenção: se o programa acima não compilar pode ser útil observar o tópico configurações que se encontra logo abaixo neste tutorial.

4- Carregando código executável para o simulador

Antes de carregar seu programa para o simulador o programa Mips.exe deverá ser executado. O carregamento para o simulador será feito através do programa MipsIt como está indicado na figura abaixo. Após o carregamento a janela do simulador é aberta automaticamente.



Devemos, neste momento, passar para a manipulação do simulador.

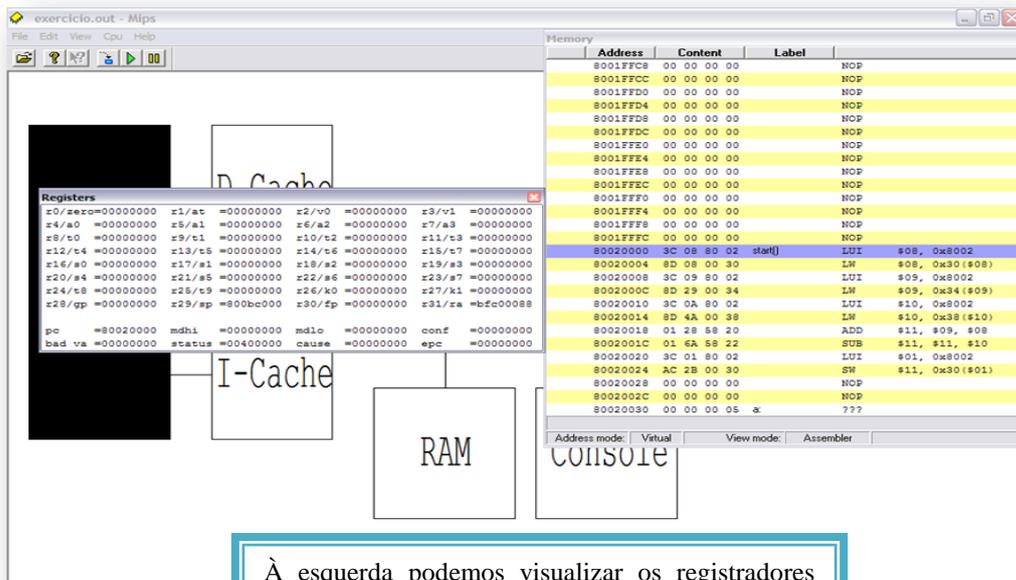
5- Usando o simulador.

Uma vez que o código foi carregado no simulador, a janela de simulação é aberta e o usuário pode verificar os registradores (clcando no desenho da CPU) ou a memória (clcando no desenho RAM). Outros componentes como memória cache e

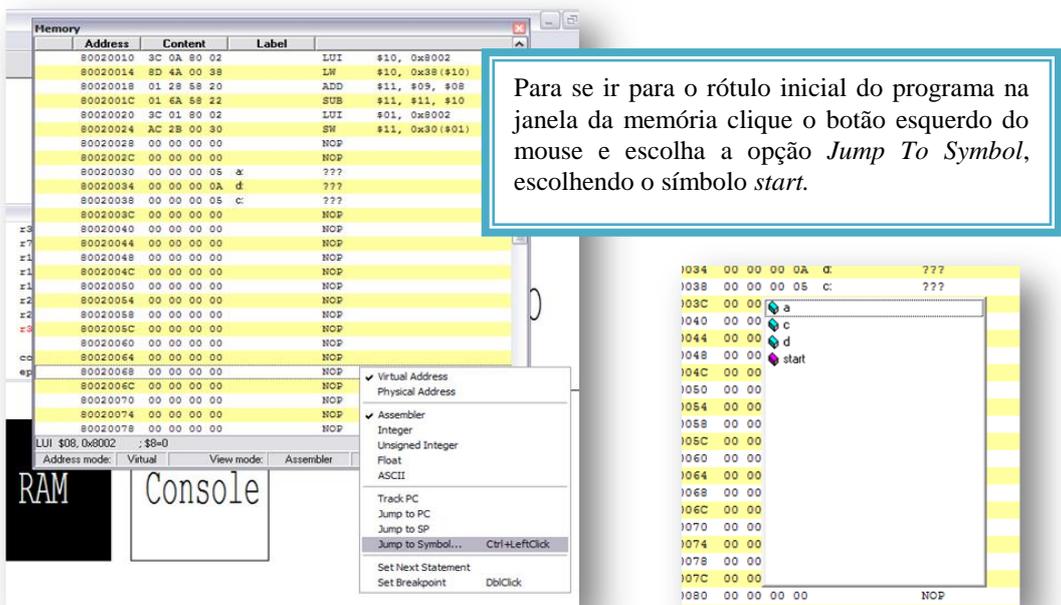
dispositivos de E/S poderão ser visualizados, mas estão fora do escopo desta etapa inicial. Clique apenas na CPU e na RAM.

O quadro de registradores mostra o conteúdo de todos os registradores em notação hexadecimal e para cada registrador tem-se o número do mesmo e o nome simbólico. Além dos registradores de propósito geral é possível visualizar alguns registradores específicos como o PC.

A visualização da memória inclui quatro colunas: endereço, conteúdo em hexadecimal, rótulos e instrução de máquina simbólica. Com isto pode-se reconhecer as instruções digitadas no programa exemplo. Veja as figuras abaixo para obter detalhes.



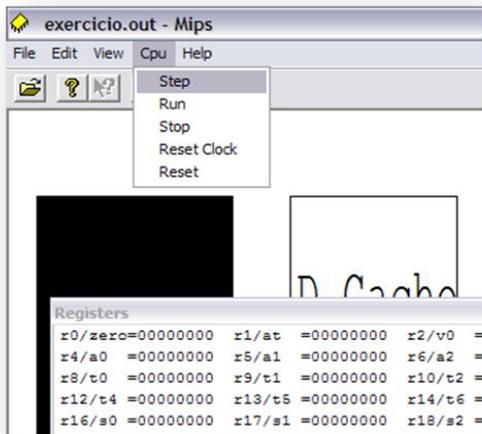
À esquerda podemos visualizar os registradores que compõem a CPU e à direita vemos a disposição da memória como no comentário acima.



Para se ir para o rótulo inicial do programa na janela da memória clique o botão esquerdo do mouse e escolha a opção *Jump To Symbol*, escolhendo o símbolo *start*.

6- Executando um programa passo a passo

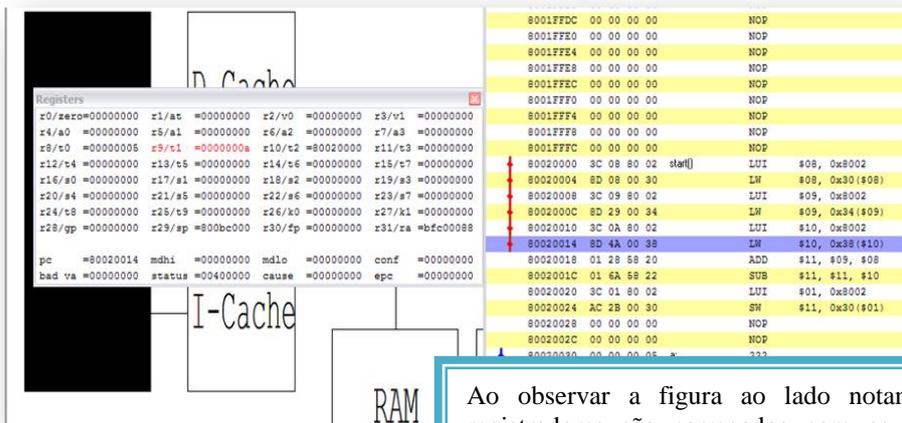
Chegou a hora de executar o programa que foi apresentado. Para tanto siga as figuras abaixo.



Após ter ido para o rótulo de início como indicado na seção anterior, escolha a opção *Step* dentro do menu *CPU* do simulador Mips.exe ou clique no botão com uma seta apontando para quadrado azul.



Após todos esses passos chegou à hora de executarmos o programa. Faça-o e veja o que acontece...



Ao observar a figura ao lado notamos que os registradores são carregados com os valores que estavam na memória e a seqüência de instruções sendo seguidas na memória

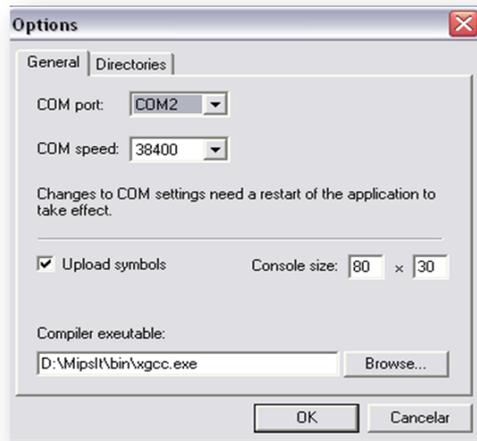
7- Usando nome simbólico dos registradores

Para poder se usar o nome simbólico dos registradores em vez de seus números inclua o texto: `include <Iregdef.h>` no início do programa. Edite o arquivo exemplo.s incluindo esta linha e use os nomes dos registradores `$8= t0`, `$9= t1` e `$10 = t2`. Salve este arquivo, recompile e simule novamente. Mude os valores das variáveis, compile e simule novamente...

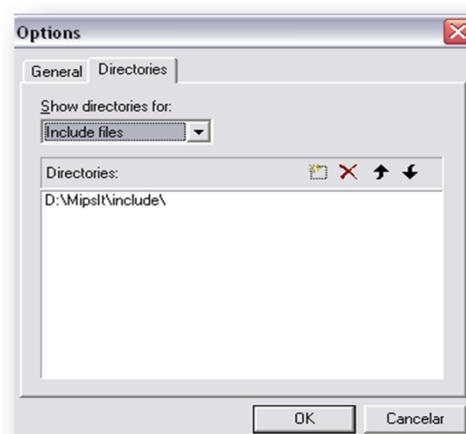
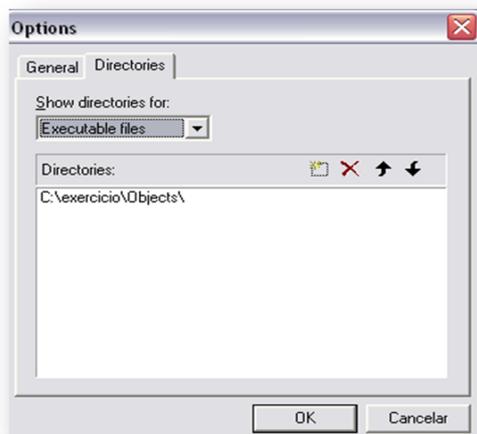
8 - Configurações

Agora vamos ver algumas configurações básicas que podem ser o diferencial no momento de compilar o programa. Em caso de alguma dúvida consulte os monitores.

Para começar, selecione o menu File do MipsIt e escolha a opção Options. Neste momento irá aparecer a janela abaixo. É nela que as alterações devem ser feitas.



Em Compiler executable indique o endereço do programa xgcc.exe. Este executável deve estar na pasta bin.



Use a opção *show directories for* para selecionar os diretórios onde estão arquivos auxiliares do MipsIt.
Siga a seguinte configuração:
Executable files aponta para a pasta Objects do MipsIt.
Include files aponta para o diretório include do MipsIt.
Library files aponta para o diretório lib do MipsIt.

Após esta breve instrução ao simulador, resolva os exercícios da lista referente a essa aula e simule os programas propostos.