

1. GCC

1.1 Caminhos de um programa:

pre-processamento → compilação → Linkagem

1.2 GCC:

gcc meu_programa.c

Todas as etapas executadas.

1.3 Diretivas:

-o) Diz qual é o nome que voce gostaria que fosse nomeado a saída.

-g) Diretiva usada para gerar, junto ao executavel, arquivos que permitem a execução do gdb e outros debuggers

-c) Permite interromper o processo de geração do programa antes da linkagem. Gerando assim objetos (.o)

2. Make

2.1 Porque usar:

– Automação do processo

– Otimização do processo de compilação (compila de novo somente oque é necessario, ou seja, foi modificado após a data corrente)

– Abstração

2.2 Estrutura:

regra: dependências

comando

comando

comando ...

3. GDB

3.1 Usando GDB:

Lembre-se sempre de usar a diretiva -g para que o programa funcione adequadamente.

Para executar o programa use o seguinte comando:

`gdb meu_programa`

Sendo “meu_programa” um executavel.

3.2 Break Points

O gdb é muito util, pois usando ele voce pode parar no meio do programa utilizando os break points. Os break points são os lugares onde o programa deve pausar sua execução, para criar use:

`break <linha>`

Assim, o programa, a chegar na linha explicitada ira pausar. Com o programa pausado voce pode ver os valores de variaveis que estao no escopo usando:

`display <variavel>`

Assim voce pode saber exatamente oque esta acontecendo com cada variavel em determinados pontos. Para continuar o programa basta usar:

`continue`

3.3 Step, next & where

O gdb, apos chegar em um breackpoint, permite você ir andando linha por linha e cada linha ver os valores das variaveis, utilizando o comando:

next

Voce estara ordando o gdb a executar a linha em que ele esta, independente do comando. Ou seja caso seja um printf, ele ira executar somente o printf, mas caso seja um chamada para uma funcao por exemplo ele ira executar a funcao toda e so depois ira pausar outra vez. Caso voce queira “entrar” dentro da funcao, o comando correto é:

step

Caso o programa se encerre abruptamente (como sigmentation fault), o gdb é capaz de dectar em que linha ocorreu este erro, basta utilizar o comando:

where

MAKE

| Tutorial ditatico : <http://orion.lcg.ufrj.br/compgraf1/downloads/MakefileTut.pdf> |

| Tutorial oficial: <http://www.opussoftware.com/tutorial/TutMakefile.htm> |

GCC

| Tutorial didatico1: <http://pages.cs.wisc.edu/~beechung/ref/gcc-intro.html>

| Tutorial didatico2: <http://forum.imasters.com.br/index.php?/topic/348057-tutorial-gcc-o-basico/> |

| Tutorial oficial: <http://gcc.gnu.org/onlinedocs/> |

GDB

| Tutorial didatico1: <http://www.cs.umd.edu/~srhuang/teaching/cmsc212/gdb-tutorial-handout.pdf>

| Tutorial didatico2: <http://jumpi.wordpress.com/2007/07/24/tutorial-basico-de-gdb/>

| Tutorial oficial: <http://www.gnu.org/software/gdb/documentation/> |
