

Uma Introdução aos Compiladores

Diogo Dutra Albuquerque
Baltazar Tavares Vanderlei

Laboratório de Computação Científica e Visualização - LCCV/UFAL

23 de Outubro de 2008

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

O que é um compilador?

“Um compilador é um programa(ou grupo de programas) que aceitando como entrada um arquivo escrito em uma linguagem (o código fonte) gera um programa em outra linguagem(o código objeto).”

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

O que é um compilador?
Como Funciona um Compilador C++?
Flags de Compilação genéricas
Flags de Compilação do g++
Flags de Compilação do icpc
Makefile
IDE
Netbeans

Pré-Processador
Compilação
Link Edição(Linkagem)

Pré-Processador

- Primeira etapa do processo de compilação em c++.

Pré-Processador

- Primeira etapa do processo de compilação em c++.
- Responsável de resolver as diretivas de pré-compilação.

Pré-Processador

- Primeira etapa do processo de compilação em c++.
- Responsável de resolver as diretivas de pré-compilação.
- No final dessa etapa, será gerado um código sem nenhum “#” do usuário.

Exemplo com o g++

pre.cpp: Arquivo com código a ser pré-processado

```
#include "my.h"
```

```
int main( int argc , char *argv[] ){  
    sample();  
    return 0;  
}
```

Exemplo com o g++

pre.cpp: Arquivo com código a ser pré-processado

```
#include "my.h"
```

```
int main( int argc , char *argv[] ){  
    sample();  
    return 0;  
}
```

my.h: Arquivo incluído em pre.cpp

```
int sample();
```

Resultado do pré-processamento com o g++

```
g++ -E samples/pre.cpp -DMSG="" Hal, open the pod doors."
# 1 "samples/pre.cpp"
# 1 "<built-in>"
# 1 "<command-line>"
# 1 "samples/pre.cpp"
# 1 "samples/my.h" 1
int sample();
# 2 "samples/pre.cpp" 2

int main( int argc , char *argv[] ){
    sample();
    return 0;
}
```

Flags uteis de Pré-Processamento

Como flags mais úteis do pré-processamento temos:

- `-I` : indica ao compilador o diretório das bibliotecas.

Flags uteis de Pré-Processamento

Como flags mais úteis do pré-processamento temos:

- -I : indica ao compilador o diretório das bibliotecas.
- -D : Equivalente a um #define

Exemplo com o g++

define.cpp: Arquivo com código a ser pré-processado

```
int main(){  
    char *c=MSG;  
    return 0;  
}
```

Resultado do pré-processamento com o g++

```
g++ -E samples/define.cpp -DMSG="" Hal, open the pod door  
# 1 "samples/define.cpp"  
# 1 "<built-in>"  
# 1 "<command-line>"  
# 1 "samples/define.cpp"  
int main(){  
    char *c="Hal, open the pod doors."  
    return 0;  
}
```


- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - **Compilação**
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

O que é feito na etapa de compilação?

- O processo de compilação em si(dããa).

O que é feito na etapa de compilação?

- O processo de compilação em si(dããa).
- A checagem de erros sintáticos é feito nessa etapa.

O que é feito na etapa de compilação?

- O processo de compilação em si(dããa).
- A checagem de erros sintáticos é feito nessa etapa.
- A geração de código binário de acordo com a maquina especificada.

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

O que é a etapa de linkagem?

- Nessa etapa serão resolvidas as dependências com bibliotecas externas.

O que é a etapa de linkagem?

- Nessa etapa serão resolvidas as dependências com bibliotecas externas.
- Será feito a ligação entre os arquivos do programa.

O que é a etapa de linkagem?

- Nessa etapa serão resolvidas as dependências com bibliotecas externas.
- Será feito a ligação entre os arquivos do programa.
- Também será resolvido as bibliotecas estáticas.

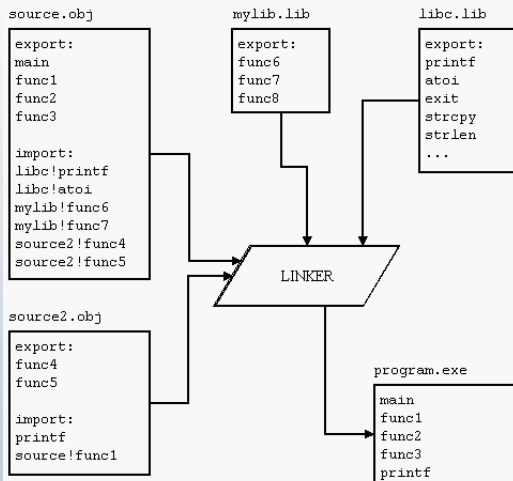
O que é a etapa de linkagem?

- Nessa etapa serão resolvidas as dependências com bibliotecas externas.
- Será feito a ligação entre os arquivos do programa.
- Também será resolvido as bibliotecas estáticas.
- E a etapa em que você mais deve ter conhecimento global do projeto.

O que é um compilador?
Como Funciona um Compilador C++?
Flags de Compilação genéricas
Flags de Compilação do g++
Flags de Compilação do icpc
Makefile
IDE
Netbeans

Pré-Processador
Compilação
Link Edição(Linkagem)

Esquema de linkagem



- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 **Flags de Compilação genéricas**
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

Os níveis de otimização On:

O que é os níveis de otimização -O[n]?

- Os níveis de otimização -O[n] são chaves para habilitar flags de otimização em diversos compiladores.

Os níveis de otimização On:

O que é os níveis de otimização -O[n]?

- Os níveis de otimização -O[n] são chaves para habilitar flags de otimização em diversos compiladores.
- Para cada nível diferente e melhor, um numero [n] maior.

Os níveis de otimização On:

O que é os níveis de otimização -O[n]?

- Os níveis de otimização -O[n] são chaves para habilitar flags de otimização em diversos compiladores.
- Para cada nível diferente e melhor, um numero [n] maior.
- Pode ir de "1" ate "5", mas não em todos os compiladores.

Os níveis de otimização On:

O que é os níveis de otimização -O[n]?

- Os níveis de otimização -O[n] são chaves para habilitar flags de otimização em diversos compiladores.
- Para cada nível diferente e melhor, um numero [n] maior.
- Pode ir de "1" ate "5", mas não em todos os compiladores.
- A única exceção é o nível "s": Ele otimiza o tamanho do executável.

Exemplos:

g++ (Compilador da GNU):

- g++ -O2 main.c

icc (Compilador da Intel®):

suncc (Compilador da SUN):

Exemplos:

g++ (Compilador da GNU):

- g++ -O2 main.c

icc (Compilador da Intel®):

- icc -O2 main.c

suncc (Compilador da SUN):

Exemplos:

g++ (Compilador da GNU):

- g++ -O2 main.c

icc (Compilador da Intel®):

- icc -O2 main.c

suncc (Compilador da SUN):

- suncc -O2 main.c

O que é um compilador?
Como Funciona um Compilador C++?
Flags de Compilação genéricas
Flags de Compilação do g++
Flags de Compilação do icpc
Makefile
IDE
Netbeans

-On
-march=ARCH
-m(32 ou 64)
-Wall

1ª Tabela de opções ativadas pelo O[n] no g++:

Optimization	Included in Level			
	-O1	-O2	-Os	-O3
gcse	○	●	●	●
expensive-optimizations	○	●	●	●
strength-reduce	○	●	●	●
rerun-cse-after-loop	○	●	●	●
rerun-loop-opt	○	●	●	●
caller-saves	○	●	●	●
force-mem	○	●	●	●
peephole2	○	●	●	●
regmove	○	●	●	●
strict-aliasing	○	●	●	●
delete-null-pointer-checks	○	●	●	●
reorder-blocks	○	●	●	●
schedule-insns	○	●	●	●
schedule-insns2	○	●	●	●
inline-functions	○	○	○	●

O que é um compilador?
 Como Funciona um Compilador C++?
Flags de Compilação genéricas
 Flags de Compilação do g++
 Flags de Compilação do icpc
 Makefile
 IDE
 Netbeans

-On
 -march=ARCH
 -m(32 ou 64)
 -Wall

2ª Tabela de opções ativadas pelo O[n] no g++:

	Included in Level			
Optimization	-O1	-O2	-Os	-O3
defer-pop	●	●	●	●
thread-jumps	●	●	●	●
branch-probabilities	●	●	●	●
cprop-registers	●	●	●	●
guess-branch-probability	●	●	●	●
omit-frame-pointer	●	●	●	●
align-loops	○	●	○	●
align-jumps	○	●	○	●
align-labels	○	●	○	●
align-functions	○	●	○	●
optimize-sibling-calls	○	●	●	●
cse-follow-jumps	○	●	●	●
cse-skip-blocks	○	●	●	●

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

-march: A flag para escolha de arquitetura.

Essa flag especifica ao compilador para qual tipo de maquina deve ser gerado o executável do seu programa. Assim o programa poder ser compilado com instruções binárias melhores para o tipo de maquina que ele ira rodar.

Algumas arquiteturas disponíveis para o g++:

i386 (Um dos processadores mais antigos da Intel®):

- g++ -march=i386 main.c

i486:

pentium3 (Modelo já com suporte a MMX e SSE):

core2 (Um dos modelos mais novos da Intel®):

athlon-4:

pentium4 (Famoso pentium4):

Algumas arquiteturas disponíveis para o g++:

i386 (Um dos processadores mais antigos da Intel®):

- g++ -march=i386 main.c

i486:

- g++ -march=i486 main.c

pentium3 (Modelo já com suporte a MMX e SSE):

core2 (Um dos modelos mais novos da Intel®):

athlon-4:

pentium4 (Famoso pentium4):

Algumas arquiteturas disponíveis para o g++:

i386 (Um dos processadores mais antigos da Intel®):

- g++ -march=i386 main.c

i486:

- g++ -march=i486 main.c

pentium3 (Modelo já com suporte a MMX e SSE):

- g++ -march=pentium3 main.c

core2 (Um dos modelos mais novos da Intel®):

athlon-4:

pentium4 (Famoso pentium4):

Algumas arquiteturas disponíveis para o g++:

i386 (Um dos processadores mais antigos da Intel®):

- g++ -march=i386 main.c

i486:

- g++ -march=i486 main.c

pentium3 (Modelo já com suporte a MMX e SSE):

- g++ -march=pentium3 main.c

core2 (Um dos modelos mais novos da Intel®):

- g++ -march=core2 main.c

athlon-4:

pentium4 (Famoso pentium4):

Algumas arquiteturas disponíveis para o g++:

i386 (Um dos processadores mais antigos da Intel®):

- g++ -march=i386 main.c

i486:

- g++ -march=i486 main.c

pentium3 (Modelo já com suporte a MMX e SSE):

- g++ -march=pentium3 main.c

core2 (Um dos modelos mais novos da Intel®):

- g++ -march=core2 main.c

athlon-4:

- g++ -march=athlon-4 main.c

pentium4 (Famoso pentium4):

Algumas arquiteturas disponíveis para o g++:

i386 (Um dos processadores mais antigos da Intel®):

- g++ -march=i386 main.c

i486:

- g++ -march=i486 main.c

pentium3 (Modelo já com suporte a MMX e SSE):

- g++ -march=pentium3 main.c

core2 (Um dos modelos mais novos da Intel®):

- g++ -march=core2 main.c

athlon-4:

- g++ -march=athlon-4 main.c

pentium4 (Famoso pentium4):

- g++ -march=pentium4 main.c

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - **-m(32 ou 64)**
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

Para que serve as flags -m64 ou -m32 ?

Essa flag especifica ao compilador para qual “arquitetura da memória” deve ser gerado o executável do seu programa. Assim o programa poder ser compilado com instruções binárias melhores para o tipo de maquina que ele ira rodar e pode usar melhor os tipos que aumentam de tamanho com 64 bits.

As opções disponíveis são: -m32 ou -m64.

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

Para que serve a flag -Wall ?

Essa flag especifica ao compilador que o nível de alarmes que ele deve passar ao usuário é o nível máximo. Nesse nível o compilador no avisa de possíveis erros de lógica e ate mesmo otimizações no código. Este é uma flag muito usada e recomendada. O ideal é que ela sempre seja habilitada no compilador.

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 **Flags de Compilação do g++**
 - -mfpmath=tipo
 - -fopenmp
- 5 **Flags de Compilação do icpc**
- 6 **Makefile**
- 7 **IDE**
- 8 **Netbeans**

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

Para que serve a flag -mfpmath ?

Essa flag especifica ao compilador se o executável ira usar uma “unidade de pontos flutuantes” 387 ou se ele ira usar as instruções SSE para as operações de ponto flutuante.

Usar as instruções SSE é mais rápido, mas necessita de uma arquitetura de processador que de suporte(A maioria).

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

Como habilitar o suporte a openmp no g++?

A flag `-fopenmp` é usada para habilitar o suporte ao openmp no g++. Com essa flag você está apto a compilar, linkar e pré-processar todo o seu código que tiver OpenMP.

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc**
- 6 Makefile
- 7 IDE
- 8 Netbeans

ICC e ICPC

- -ip : Habilita otimização interprocedural, mas somente por arquivo

ICC e ICPC

- -ip : Habilita otimização interprocedural, mas somente por arquivo
- -ipo : Habilita otimização interprocedural multi-arquivos

ICC e ICPC

- -ip : Habilita otimização interprocedural, mas somente por arquivo
- -ipo : Habilita otimização interprocedural multi-arquivos
- -fast : Habilita as opções: -O3, -ipo, -static, -no-prec-div, e -xP

ICC e ICPC

- -ip : Habilita otimização interprocedural, mas somente por arquivo
- -ipo : Habilita otimização interprocedural multi-arquivos
- -fast : Habilita as opções: -O3, -ipo, -static, -no-prec-div, e -xP
- -no-prec-div : Desabilita precisão na divisão

ICC e ICPC

- -ip : Habilita otimização interprocedural, mas somente por arquivo
- -ipo : Habilita otimização interprocedural multi-arquivos
- -fast : Habilita as opções: -O3, -ipo, -static, -no-prec-div, e -xP
- -no-prec-div : Desabilita precisão na divisão
- -xP : Otimiza para arquiteturas acima do pentium4

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile**
- 7 IDE
- 8 Netbeans

O que é um compilador?
Como Funciona um Compilador C++?
Flags de Compilação genéricas
Flags de Compilação do g++
Flags de Compilação do icpc
Makefile
IDE
Netbeans

Makefile

make

Makefile

make

- Programa para automação e agilização de atividades

Makefile

make

- Programa para automação e agilização de atividades
- Detecta quais arquivos foram alterados

Makefile

make

- Programa para automação e agilização de atividades
- Detecta quais arquivos foram alterados

Makefile

Makefile

make

- Programa para automação e agilização de atividades
- Detecta quais arquivos foram alterados

Makefile

- Arquivo de configuração para o programa make

Makefile

make

- Programa para automação e agilização de atividades
- Detecta quais arquivos foram alterados

Makefile

- Arquivo de configuração para o programa make
- Possui uma sintaxe simples

Makefile

make

- Programa para automação e agilização de atividades
- Detecta quais arquivos foram alterados

Makefile

- Arquivo de configuração para o programa make
- Possui uma sintaxe simples
- Fácil utilização e manuseio

Makefile

make

- Programa para automação e agilização de atividades
- Detecta quais arquivos foram alterados

Makefile

- Arquivo de configuração para o programa make
- Possui uma sintaxe simples
- Fácil utilização e manuseio
- Baseia-se em regras

Makefile

make

- Programa para automação e agilização de atividades
- Detecta quais arquivos foram alterados

Makefile

- Arquivo de configuração para o programa make
- Possui uma sintaxe simples
- Fácil utilização e manuseio
- Baseia-se em regras
- As regras possuem comandos

Makefile

- Obrigatoriamente os comandos devem ser iniciados com um TAB

Makefile

- Obrigatoriamente os comandos devem ser iniciados com um TAB
regra1:
 comando_x
 comando_y

Makefile

- Obrigatoriamente os comandos devem ser iniciados com um TAB
regra1:
 comando_x
 comando_y
- A primeira regra do arquivo é a regra a ser executada por padrão

Makefile

- Obrigatoriamente os comandos devem ser iniciados com um TAB

regra1:

comando_x

comando_y

- A primeira regra do arquivo é a regra a ser executada por padrão
- As regras podem ter dependências de outras regras ou arquivos

Makefile

- Obrigatoriamente os comandos devem ser iniciados com um TAB

regra1:

comando_x

comando_y

- A primeira regra do arquivo é a regra a ser executada por padrão
- As regras podem ter dependências de outras regras ou arquivos
- Caso a dependência é uma regra, deve seguir uma hierarquia

Makefile

- Obrigatoriamente os comandos devem ser iniciados com um TAB
regra1:

comando_x

comando_y

- A primeira regra do arquivo é a regra a ser executada por padrão
- As regras podem ter dependências de outras regras ou arquivos
- Caso a dependência é uma regra, deve seguir uma hierarquia
- Sempre depender de uma regra mais abaixo

Makefile

- Obrigatoriamente os comandos devem ser iniciados com um TAB

regra1:

comando_x

comando_y

- A primeira regra do arquivo é a regra a ser executada por padrão
- As regras podem ter dependências de outras regras ou arquivos
- Caso a dependência é uma regra, deve seguir uma hierarquia
- Sempre depender de uma regra mais abaixo

regra1: regra2

comando_x

comando_y

regra2:

comando_z

Makefile

- Podem ser usadas variáveis

Makefile

- Podem ser usadas variáveis
- A declaração de uma variável deve obrigatoriamente ter um valor

Makefile

- Podem ser usadas variáveis
- A declaração de uma variável deve obrigatoriamente ter um valor
- O acesso a uma variável deve ser feito por \$(variavel)

Makefile

- Podem ser usadas variáveis
- A declaração de uma variável deve obrigatoriamente ter um valor
- O acesso a uma variável deve ser feito por \$(variavel)

COMANDO=comando_z

regra1:
\$(COMANDO)

Makefile

- Podem ser usadas variáveis
- A declaração de uma variável deve obrigatoriamente ter um valor
- O acesso a uma variável deve ser feito por \$(variavel)

COMANDO=comando_z

regra1:

\$(COMANDO)

- Podem ser declaradas variáveis ao chamar o programa make

Makefile

- Podem ser usadas variáveis
- A declaração de uma variável deve obrigatoriamente ter um valor
- O acesso a uma variável deve ser feito por \$(variavel)

COMANDO=comando_z

regra1:

\$(COMANDO)

- Podem ser declaradas variáveis ao chamar o programa make

\$ make COMANDO=comando_z

Exemplo simples de Makefile

```
CXX=g++  
CXXFLAGS=-O2 -pipe  
LDFLAGS=-L/usr/lib/hello -lhello  
CC=$(CXX) $(CXXFLAGS) $(LDFLAGS)  
  
all: hello hello.o main.c  
    $(CC) main.c hello.o -o hello  
  
hello: hello.c hello.h  
    $(CXX) -c hello.c
```

- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

O que é um compilador?
Como Funciona um Compilador C++?
Flags de Compilação genéricas
Flags de Compilação do g++
Flags de Compilação do icpc
Makefile
IDE
Netbeans

IDE

- Integrated Development Environment - Ambiente de desenvolvimento integrado

IDE

- Integrated Development Environment - Ambiente de desenvolvimento integrado
- Uma IDE é um conjunto de várias ferramentas

IDE

- Integrated Development Environment - Ambiente de desenvolvimento integrado
- Uma IDE é um conjunto de várias ferramentas

As ferramentas mais comuns de se encontrar em uma IDE são:

- Editor de texto

IDE

- Integrated Development Environment - Ambiente de desenvolvimento integrado
- Uma IDE é um conjunto de várias ferramentas

As ferramentas mais comuns de se encontrar em uma IDE são:

- Editor de texto
- Compilador

IDE

- Integrated Development Environment - Ambiente de desenvolvimento integrado
- Uma IDE é um conjunto de várias ferramentas

As ferramentas mais comuns de se encontrar em uma IDE são:

- Editor de texto
- Compilador
- Depurador

IDE

- Integrated Development Environment - Ambiente de desenvolvimento integrado
- Uma IDE é um conjunto de várias ferramentas

As ferramentas mais comuns de se encontrar em uma IDE são:

- Editor de texto
- Compilador
- Depurador
- Normalmente as IDEs têm dois modos de compilação, Debug e Release

IDE

- Integrated Development Environment - Ambiente de desenvolvimento integrado
- Uma IDE é um conjunto de várias ferramentas

As ferramentas mais comuns de se encontrar em uma IDE são:

- Editor de texto
- Compilador
- Depurador
- Normalmente as IDEs têm dois modos de compilação, Debug e Release
- Existem várias IDEs. Alguns exemplos são Netbeans, Eclipse, Geany, Anjuta, CodeBlocks...

O que é um compilador?
Como Funciona um Compilador C++?
Flags de Compilação genéricas
Flags de Compilação do g++
Flags de Compilação do icpc
Makefile
IDE
Netbeans

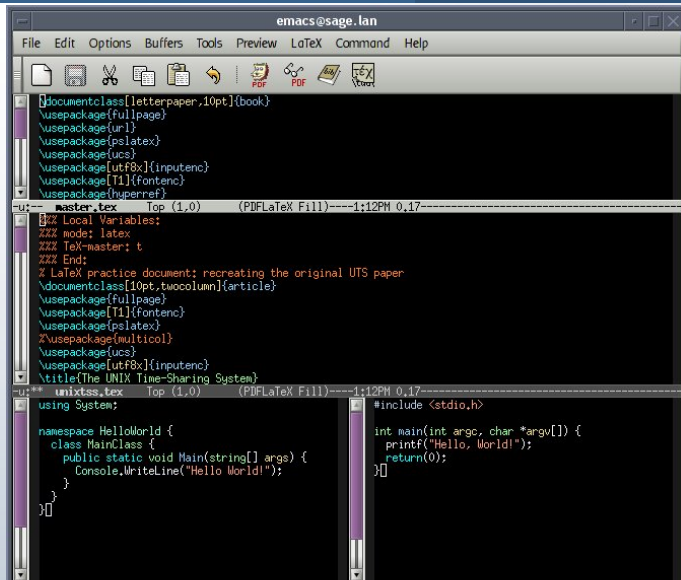
The screenshot displays the NetBeans IDE interface with the following components:

- Menu Bar:** File, Edit, View, Goto, Project, Build, Tools, Debug, CVS, Settings, Help.
- Toolbar:** Standard IDE icons for file operations and development.
- Symbols Panel (Left):**
 - Tree:** Shows a hierarchy of classes under 'Classes'. The 'CallTip' class is expanded, showing methods like `CallTip(const Ca`, `CallTip()`, `CallTipCancel()`, `CallTipStart(int p`, `DrawChunk(Surf`, `MouseClicked(Poin`, `PaintCT(Surface`, `PaintContents(Si`, and `RefreshColourPa`.
 - Search:** Empty search field.
- Inheritance Graph (Center):** A diagram showing the relationships between classes:
 - `DocWatcher` and `LexerModule` inherit from `Editor`.
 - `Editor` inherits from `ScintillaBase`.
 - `ScintillaBase` inherits from `ScintillaGTK`.
 - `Font` inherits from `FontCached`.
 - `Window` inherits from `ListBox`.
 - `ListBox` inherits from `ListBoxX`.
 - `wxApp` inherits from `MyApp`.
 - `ExternLexModule` is highlighted in blue, indicating it is the selected class.
- Project Explorer (Right):** Shows the project structure:
 - `libanjuta-interfaces.la`
 - `libanjuta.la`
 - `libanjuta-egg.la`
 - `test-multi-drag`
 - `test-tree-utils`
 - `test-actions` (expanded):
 - `test-actions.c`
 - `test-union`
- Files Panel (Right):** Shows the file structure of the `anjuta` project:
 - `anjuta` (expanded):
 - `autom4te.cache`
 - `CVS`
 - `data`
 - `doc`
 - `global-tags`
 - `launcher`
 - `libanjuta`
- Terminal (Bottom):** Shows the command prompt output:

```
[naba@localhost naba]$ ls
accounts  belzabar  Documents  evolution  hexagon.png  logo  Movies  rpms  vnc-scripts
anjuta    Desktop  Downloads  GNUstep   Images       mail  Projects  test.txt
[naba@localhost naba]$
```

At the bottom of the IDE, the status bar shows: Col: 000 Line: 0117 Project: anjuta Mode: INS Zoom: 0.

O que é um compilador?
Como Funciona um Compilador C++?
Flags de Compilação genéricas
Flags de Compilação do g++
Flags de Compilação do icpc
Makefile
IDE
Netbeans



```
emacs@sage.lan
File Edit Options Buffers Tools Preview LaTeX Command Help

\documentclass[letterpaper,10pt]{book}
\usepackage{fullpage}
\usepackage{url}
\usepackage{pslatex}
\usepackage{ucs}
\usepackage[utf8x]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{hyperref}

-- master.tex Top (1,0) (PDFLaTeX Fill) --- 1:12PM 0.17 ---
%% Local Variables:
%% mode: latex
%% TeX-master: t
%% End:
% LaTeX practice document: recreating the original UTS paper
\documentclass[10pt,twocolumn]{article}
\usepackage{fullpage}
\usepackage[T1]{fontenc}
\usepackage{pslatex}
\usepackage{multicol}
\usepackage{ucs}
\usepackage[utf8x]{inputenc}
\title{The UNIX Time-Sharing System}

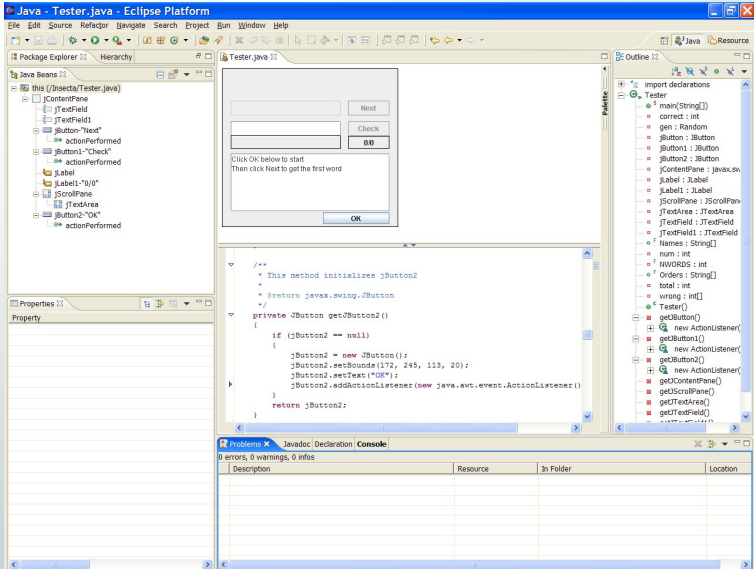
-- unixts.tex Top (1,0) (PDFLaTeX Fill) --- 1:12PM 0.17 ---
using System;

namespace HelloWorld {
    class MainClass {
        public static void Main(string[] args) {
            Console.WriteLine("Hello World!");
        }
    }
}

#include <stdio.h>

int main(int argc, char *argv[]) {
    printf("Hello, World!");
    return(0);
}
```

O que é um compilador?
Como Funciona um Compilador C++?
Flags de Compilação genéricas
Flags de Compilação do g++
Flags de Compilação do icpc
Makefile
IDE
Netbeans



- 1 O que é um compilador?
- 2 Como Funciona um Compilador C++?
 - Pré-Processador
 - Compilação
 - Link Edição(Linkagem)
- 3 Flags de Compilação genéricas
 - -On
 - -march=ARCH
 - -m(32 ou 64)
 - -Wall
- 4 Flags de Compilação do g++
 - -mfpmath=tipo
 - -fopenmp
- 5 Flags de Compilação do icpc
- 6 Makefile
- 7 IDE
- 8 Netbeans

O que é um compilador?
Como Funciona um Compilador C++?
Flags de Compilação genéricas
Flags de Compilação do g++
Flags de Compilação do icpc
Makefile
IDE
Netbeans

Netbeans

- IDE de fácil usabilidade

O que é um compilador?
Como Funciona um Compilador C++?
Flags de Compilação genéricas
Flags de Compilação do g++
Flags de Compilação do icpc
Makefile
IDE
Netbeans

Netbeans

- IDE de fácil usabilidade
- Multiplataforma

O que é um compilador?
Como Funciona um Compilador C++?
Flags de Compilação genéricas
Flags de Compilação do g++
Flags de Compilação do icpc
Makefile
IDE
Netbeans

Netbeans

- IDE de fácil usabilidade
- Multiplataforma
- Suporte a plugins

Netbeans

- IDE de fácil usabilidade
- Multiplataforma
- Suporte a plugins
- Vastamente utilizada pelo mundo

Netbeans

- IDE de fácil usabilidade
- Multiplataforma
- Suporte a plugins
- Vastamente utilizada pelo mundo
- Recomendado o uso pelo Curso de C++ LCCV