

Técnicas para desenvolvimento e aceleração de códigos científicos

Raul Baldin
LabMeC - FEC
UNICAMP

**Minicurso
LNCC 2014**

Agenda

- Controle de versão
- SVN
- SVN - Exemplo
- Processo de build
- CMake
- CMake - Exemplo

Controle de versão

Motivação

- **O que é um controle de versão?**

É a tarefa de controlar e adicionar uma numeração única à um estado específico de um documento ou conjunto de documentos.

- **Por que usar um sistema de controle de versão?**

- Fornece um histórico de mudanças
- Fornece segurança de dados
- Facilita colaboração na edição de artigos, documentos, códigos, etc.

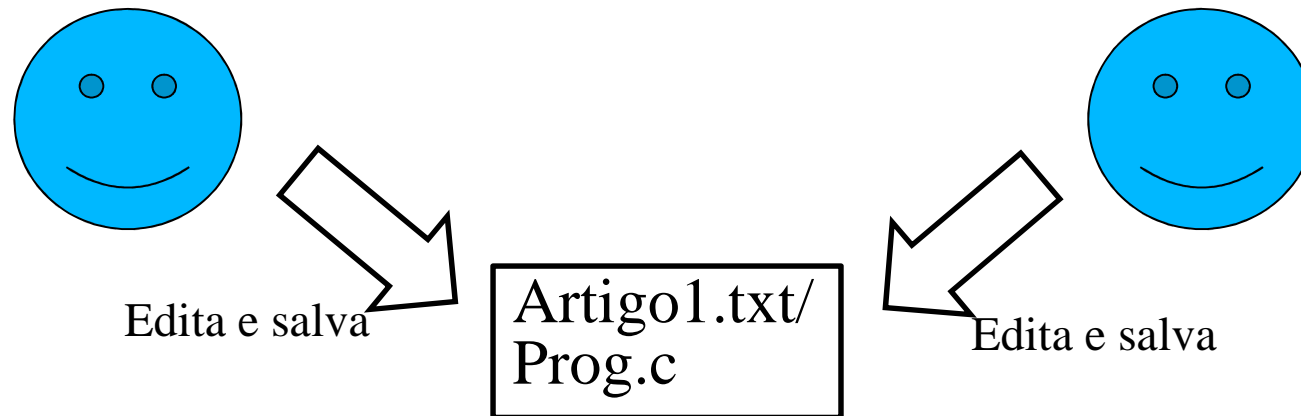
Controle de versão

Motivação

- **Cenário 1:**

Arquivo compartilhado em pasta compartilhada

- Qualquer um pode editar o arquivo
- Pode sobrescrever modificações anteriores



Controle de versão

Motivação

- **Cenário 2:**

Pedido de uma revisão de artigo por e-mail

- Após enviar para revisão, você encontra alguns erros, ou deseja reescrever certa parte
- Necessidade de aguardar o arquivo voltar da revisão
- Ou alterar sua cópia local e depois relembrar e integrar as modificações à cópia revisada

Controle de versão

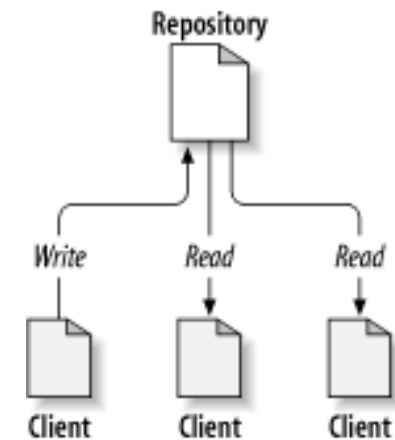
Como funciona?

Servidor:

- Central onde dados são armazenados
- Sistema de arquivos (árvore)

Clientes:

- Envia suas contribuições
- Recebem contribuições dos outros



<http://svnbook.red-bean.com/>

Controle de versão

Como funciona?

- Isso não é o conceito de um servidor de arquivos?

Controle de versão

Como funciona?

- SIM, é um tipo de servidor de arquivos!

MAS possui características interessantes!

Controle de versão

Como funciona?

- Não duplica arquivos
 - Armazena só as mudanças realizadas
 - Economiza espaço em disco
- Gerencia mudanças
 - Diminuindo retrabalho
- Permite resgatar estados (versões) anteriores de seus arquivos / diretórios

Controle de versão

Como funciona?

- Responde questões sobre o histórico de mudanças:
 - Quem foi o último que alterou um arquivo?
 - O que tinha nesse diretório 10 dias atrás?
 - Quais as diferenças entre a minha cópia e a de ontem?
 - Quem apagou um arquivo? Consigo recuperá-lo?

Agenda

- Controle de versão
- SVN
- SVN - Exemplo
- Processo de build
- CMake
- CMake - Exemplo

SVN – O que é?

- SVN → abreviação de *Subversion*
- Sistema/ferramenta de controle de versões
- *Free/open-source*
- Permite que várias pessoas em diferentes locais modifiquem o mesmo conjunto de arquivos

SVN – O que é?

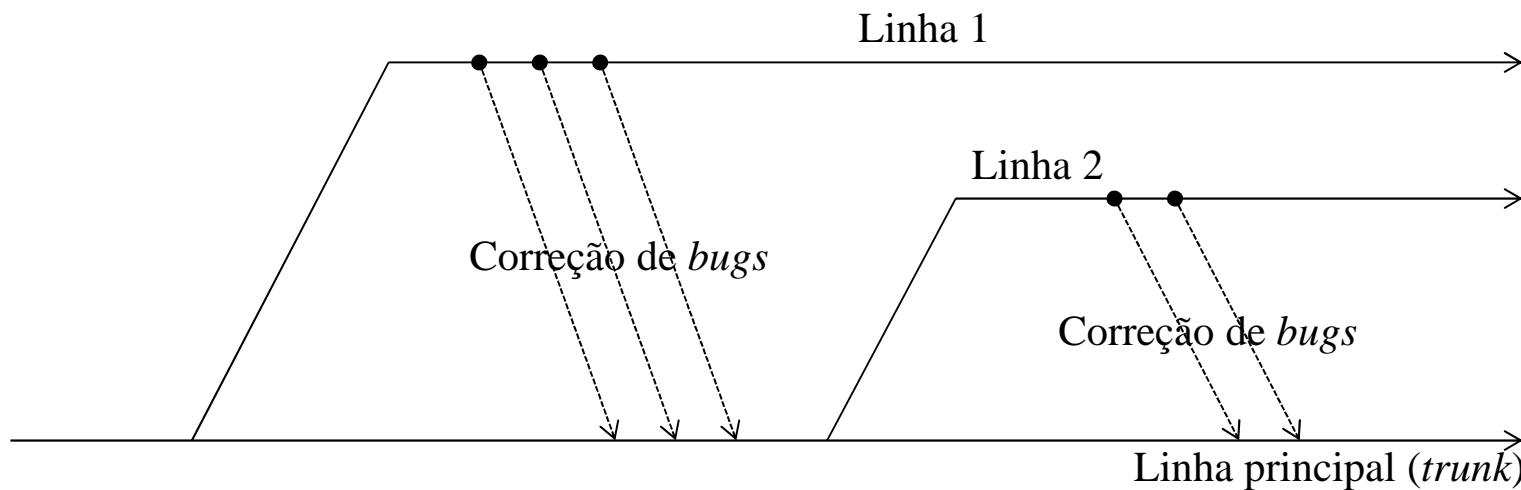
- Gerencia arquivos/diretórios e suas modificações ao longo do tempo
- Permite examinar o histórico de mudanças
- Permite recuperar versões antigas
- É uma "máquina do tempo" para arquivos :)

SVN – O que é?

- Permite trabalhar *offline* (armazena uma cópia local do repositório)
- Necessita conexão nas operações sobre o repositório (*update, commit*)
- Funciona com outros tipos de arquivos, além de fontes (cpp, latex, html, **fortran**, ..) e executáveis

SVN – O que é?

- Possibilidade de criar diferentes linhas de desenvolvimento (*branches*)
- Possibilidade de mesclar essas linhas paralelas com a linha principal (*trunk*)



SVN – Fluxo de trabalho

- | | |
|--|--|
| 1) Baixar / atualizar cópia local | 1) <code>svn checkout/update</code> |
| 2) Modificar arquivos | 2) <code>svn add/delete/copy/move</code> |
| 3) Examinar suas mudanças | 3) <code>svn status/diff</code> |
| 4) Reverter algumas mudanças | 4) <code>svn revert</code> |
| 5) Mesclar modificações / Resolver conflitos | 5) <code>svn update/resolve</code> |
| 6) Enviar suas modificações | 6) <code>svn commit</code> |

SVN – Criar repositório

- Google codes (free)

<http://code.google.com/hosting/createProject>

- Servidor Linux

<http://svnbook.red-bean.com/en/1.7/svn.reposadmin.create.html>

Create a new project

Instantly create your open source hosting project by filling out the form below. For your project, you'll receive:

- Git, Mercurial, and Subversion code hosting
- Download/release hosting
- Integrated source code browsing and code review tools
- An issue tracker and project wiki



[Learn more](#)

Project name

Example: my-project-name

Project summary

Description

Version control system
 Git
 Mercurial
 Subversion

Source code license

Project label(s)

Academic	Training	Test
<input type="text"/>	<input type="text"/>	<input type="text"/>

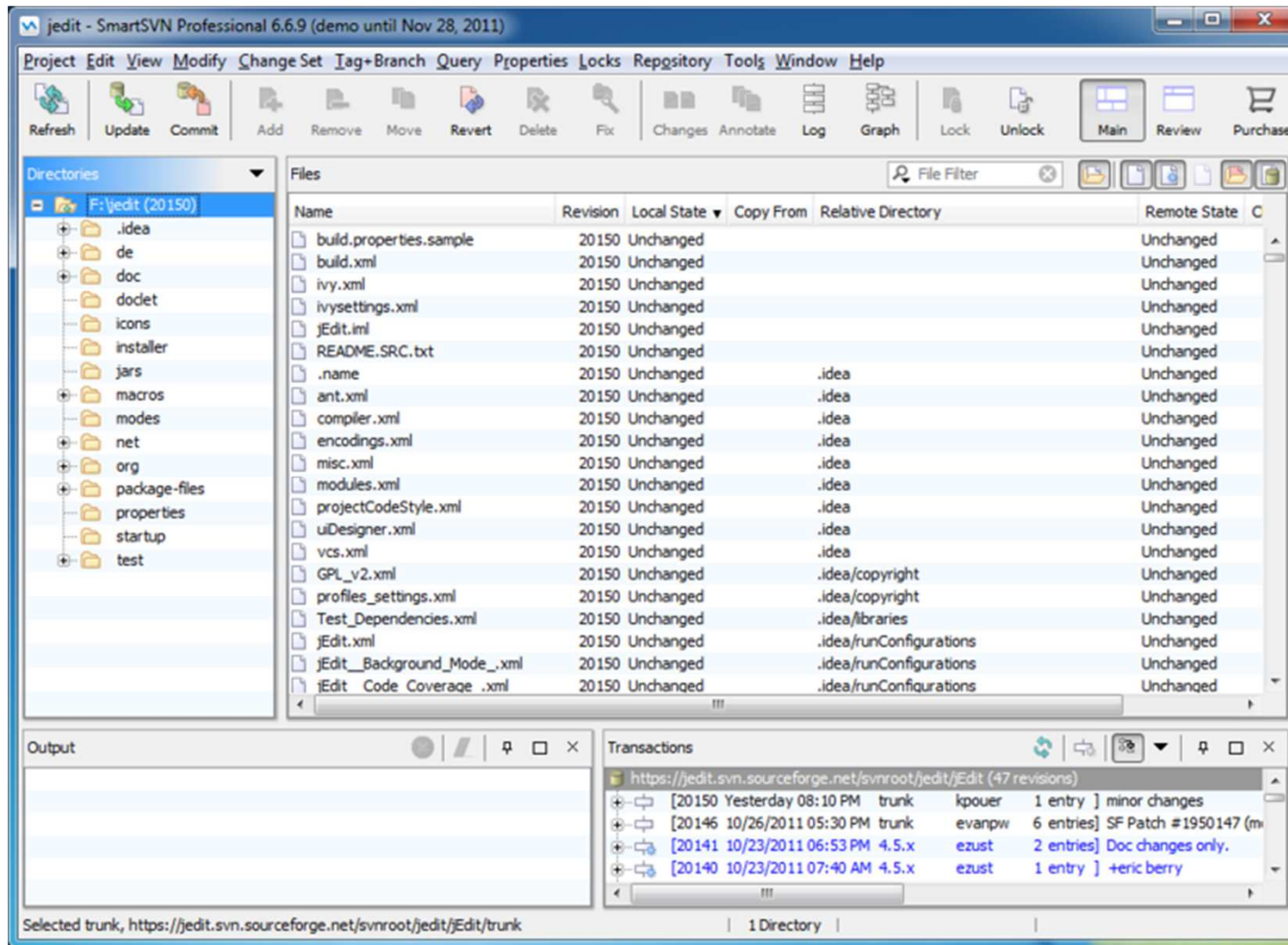
[add another row](#)

Imagem: code.google.com

SVN - Interfaces

- Existem diversas interfaces gráficas disponíveis
 - TortoiseSVN
 - SmartSVN
- Facilidade de uso
- Não precisa decorar os comandos
- *Plugins* e integrações com ambientes de desenvolvimento

SVN - Interfaces



SVN - Comandos

Salvar/Baixar arquivos (checkout)

Sintaxe: \$ svn checkout URL CAMINHO

Envio de de modificações (commit)

Sintaxe: \$ svn commit -m "Mensagem de log"

Atualização de arquivos / revisões (update)

Sintaxe: \$ svn update CAMINHO

SVN - Comandos

Listar modificações (status)

Sintaxe: `$ svn status`

Mostra diferenças entre versões (diff)

Sintaxe: `$ svn diff ARQUIVO`

`$ svn -r R1:R2 diff ARQUIVO`

Log de modificações (log)

Sintaxe: `$ svn log PATH`

SVN - Comandos

Adicionar arquivos (add)

Sintaxe: \$ svn add ARQUIVO_OU_DIRETORIO

Remover arquivos (delete*)

Sintaxe: \$ svn delete ARQUIVO_OU_DIRETORIO

Mover arquivos (move*)

Sintaxe: \$ svn move ORIGEM DESTINO

Resolvendo conflitos

```
$ # Resolver o conflito da linha manualmente,  
$ # Utilizando um editor de textos qualquer  
$ # Depois marcar o arquivo como resolvido  
$ svn resolved ARQUIVO
```

* mantém histórico de modificações

Agenda

- Controle de versão
- SVN
- **SVN - Exemplo**
- Processo de build
- CMake
- CMake - Exemplo

SVN - Exemplo

- Baixando repositório (1ª vez)

```
$ svn checkout  
https://lncc2014.googlecode.com/svn/trunk/ lncc2014  
Checked out revision 1.
```

```
$ cd lncc2014/
```

```
$ svn update  
Updating '.':  
At revision 1.
```

SVN - Exemplo

- Adicionando um arquivo

```
$ # Editando o paper
$ vi paper.txt
```

```
$ svn add paper.txt
A          paper.txt
```

- Enviando modificações

```
$ svn commit -m "Incluindo Paper"
Password for 'email@gmail.com':
Adding          paper.txt
Transmitting file data .
Committed revision 2.
```

SVN - Exemplo

- Adicionando um diretório

```
$ mkdir projeto1
```

```
$ svn add projeto1  
A      projeto1
```

- Enviando modificações

```
$ svn commit -m "Diretório incluído"  
Adding      projeto1  
Committed revision 3.
```

SVN - Exemplo

- Atualizar e mesclar modificações

```
$ svn update .  
Updating '.':  
A projeto2  
A projeto2/ementa.txt  
U paper.txt  
Updated to revision 5.
```

SVN - Exemplo

- Examinando mudanças (estado) do repositório local

```
$ svn status  
M      paper.txt
```

SVN - Exemplo

- Pessoas podem editar mesmo arquivo: mescla é feita automaticamente
- O que acontece quando 2 pessoas editam a mesma linha? → Conflito
- Resolvendo conflito

```
$ svn update
Updating '.':
C    paper.txt
Updated to revision 6.
Summary of conflicts:
  Text conflicts: 1
```

SVN - Exemplo

```
$cat paper.txt  
(...)  
<<<<<<< .mine  
1 Mexi também na linha 1 do capítulo 1  
=====  
1 Mexi na linha 1 (usuário 2)  
>>>>>>> .r6  
(...)
```

SVN - Exemplo

```
$ # Resolver conflito manualmente (editando
$ #arquivo e removendo ruído inserido nele)
$ vi paper.txt

$ cat paper.txt
(...)
1 Editamos a linha 1 do capitulo 1 (usuário 1 e 2)
(...)

$ svn resolved paper.txt
Resolved conflicted state of 'paper.txt'

$ svn commit -m "Nova atualização do paper"
Sending          paper.txt
Transmitting file data .
Committed revision 7.
```


SVN - Exemplo

- Examinando histórico

```
$ svn log
r6 | raulbaldin@gmail.com | 2014-01-22 16:15:08
(Wed, 22 Jan 2014) | 1 line
Usuario 2 mexeu na linha 1 do cap 1
r5 | raulbaldin@gmail.com | 2014-01-22 15:59:18
(Wed, 22 Jan 2014) | 1 line
Titulo do paper modificado
r4 | raulbaldin@gmail.com | 2014-01-22 15:54:45
(Wed, 22 Jan 2014) | 1 line
Ementa do projeto 2 adicionada
(...)
```

Agenda

- Controle de versão
- SVN
- SVN - Exemplo
- **Processo de build**
- CMake
- CMake - Exemplo

Processo de build

Etapas

1. Compilar um código fonte → gerar binário
2. Rodar testes → gerar relatório/site com resultados
3. Empacotar o binário → gerar pacote ou instalador

Processo de build

- Compilar um projeto pequeno é relativamente simples!
- Algumas linhas fazem a tarefa:
- Exemplos:

```
gcc -o exec fonte.cpp
```

```
gcc -o exec fonte.cpp -lblas
```

```
gcc -o exec fonte.cpp -I/usr/local/include
```

Processo de build

- Compilar um projeto manualmente é desafiador!
- Mais bibliotecas = maiores os comandos
- Fontes em diferentes locais = maiores os comandos
- Exemplo:

```
gcc -DPZSOURCEDIR=\"/local/neopz\" -DREFPATTERNDIR=\"/local/neopz/Refine/RefPatterns\" -DREALdouble -DSTATEdouble -I/local/neopz/Python -I/local/neopz/Util -I/local/neopz/PerfUtil -I/local/neopz/Common -I/local/neopz/Save -I/local/neopz/Matrix -I/local/neopz/Topology -I/local/neopz/Geom -I/local/neopz/SpecialMaps -I/local/neopz/Refine -I/local/neopz/Shape -I/local/neopz/Material -I/local/neopz/Material/REAL -I/local/neopz/Material/REAL/Plasticity -I/local/neopz/Material/Complex -I/local/neopz/Multigrid -I/local/neopz/Mesh -I/local/neopz/Mesh/REAL -I/local/neopz/StrMatrix -I/local/neopz/Integral -I/local/neopz/Frontal -I/local/neopz/Pre -I/local/neopz/Post -I/local/neopz/Analysis -I/local/neopz/SubStruct -I/local/neopz/LinearSolvers -I/local/neopz/External -I/local/neopz/External/sloan -I/usr/local/include -I/local/neopz_teste_cmake/Common -o pzcheckrestraint.cpp.o -c /local/neopz/Mesh/pzcheckrestraint.cpp
```

Processo de build

- Exemplo (cont.):

```
/usr/bin/ar cr libpz.a ./Analysis/pzanalysiserror.cpp.o ./Analysis/pzeuleranalysis.cpp.o
./Analysis/pzblackoilanalysis.cpp.o ./Analysis/tpzbiharmonicestimator.cpp.o ./Analysis/pzmganalysis.cpp.o
./Analysis/pzsmanal.cpp.o ./Analysis/pzanalysis.cpp.o ./Analysis/pznonlinanalysis.cpp.o
./Analysis/pzsmfrontalanal.cpp.o ./Analysis/TPZNLMultGridAnalysis.cpp.o
./Analysis/pztransientanalysis.cpp.o ./Common/pz_pthread.cpp.o ./Common/pzreal.cpp.o
./Common/TPZGuiInterface.cpp.o ./External/pzrenumbering.cpp.o ./External/pzmetis.cpp.o
./External/TPZBoostGraph.cpp.o ./External/rcm.cpp.o ./External/tpznodesetcompute.cpp.o
./External/pzsloan.cpp.o ./External/sloan/vsrtpl.cpp.o ./External/sloan/isorti.cpp.o
./External/sloan/label.cpp.o ./External/sloan/iorele.cpp.o ./External/sloan/gegra.cpp.o
./External/sloan/diamtr.cpp.o ./External/sloan/rootls.cpp.o ./External/sloan/number.cpp.o
./External/sloan/profil.cpp.o ./Frontal/TPZParFrontMatrix.cpp.o ./Frontal/TPZStackEqnStorage.cpp.o
./Frontal/TPZFrontMatrix.cpp.o ./Frontal/TPZFrontNonSym.cpp.o ./Frontal/pzvecfrontal.cpp.o
./Frontal/TPZFront.cpp.o ./Frontal/TPZFrontSym.cpp.o ./Frontal/tpzeqnarray.cpp.o
./Frontal/TPZFileEqnStorage.cpp.o ./Geom/TPZGeoCube.cpp.o ./Geom/pzgeotetrahedra.cpp.o
./Geom/TPZGeoLinear.cpp.o ./Geom/pzgeotriangle.cpp.o ./Geom/pzgeopyramid.cpp.o ./Geom/tpzgeoblend.cpp.o
./Geom/pzgeopoint.cpp.o ./Geom/pznoderep.cpp.o ./Geom/pzgeoquad.cpp.o ./Geom/pzgeoprism.cpp.o
./Integral/tpzintrule3d.cpp.o ./Integral/tpzintrulep3d.cpp.o ./Integral/tpzintrulelet.cpp.o
./Integral/pzquad.cpp.o ./Integral/adapt.cpp.o ./Integral/tpzintrulelist.cpp.o
./Integral/tpzrinteg.cpp.o ./Integral/tpzgaussrule.cpp.o ./Material/TPZReynoldsFlow.cpp.o
./Material/pzmaterial.cpp.o ./Material/tpzoutofrange.cpp.o ./Material/pzmat2dlin.cpp.o
./Material/pzuncouplemultiphysics.cpp.o ./Material/pzmaterialdata.cpp.o ./Material/pzconslaw.cpp.o
...
...
...
Continua por muitas páginas!
```

Processo de build

- Os exemplos foram feitos usando **gcc/Linux**
- Compartilhamos esse mesmo código com amigos que utilizam Windows / MACOSX, problemas:
 - Refazer manualmente os comandos para cada ambiente
 - Onde estão as bibliotecas em cada sistema?
 - Como é a sintaxe / parâmetros de cada compilador?

Processo de build

- Como resolver essa questão?
- Usando ferramentas que gerenciam as builds
- Ferramentas Multi-plataforma

- Exemplo: CMake

Agenda

- Controle de versão
- SVN
- SVN - Exemplo
- Processo de build
- **CMake**
- CMake - Exemplo

CMake

- É uma plataforma sofisticada para sistema de builds
- Desenvolvido pela Kitware (ITK, VTK, ...) em 2000
- Vê o projeto como um sistema de arquivos
- Suporta estruturas de diretórios complexas
- Linguagem simplificada
- É multi-plataforma (*Windows, Linux, MACOSX, etc*)
- É *open-source*
- Cria “projetos” para *MS Visual Studio, Apple XCode, Kdevelop, Borland, CodeBlocks, Eclipse* e **Makefiles**
- Funciona com tipos de compiladores (icc, gcc, msvc, **gfortran**, etc)
- Possui interface gráfica p/ múltiplos ambientes

CMake

O que realmente é o CMake?

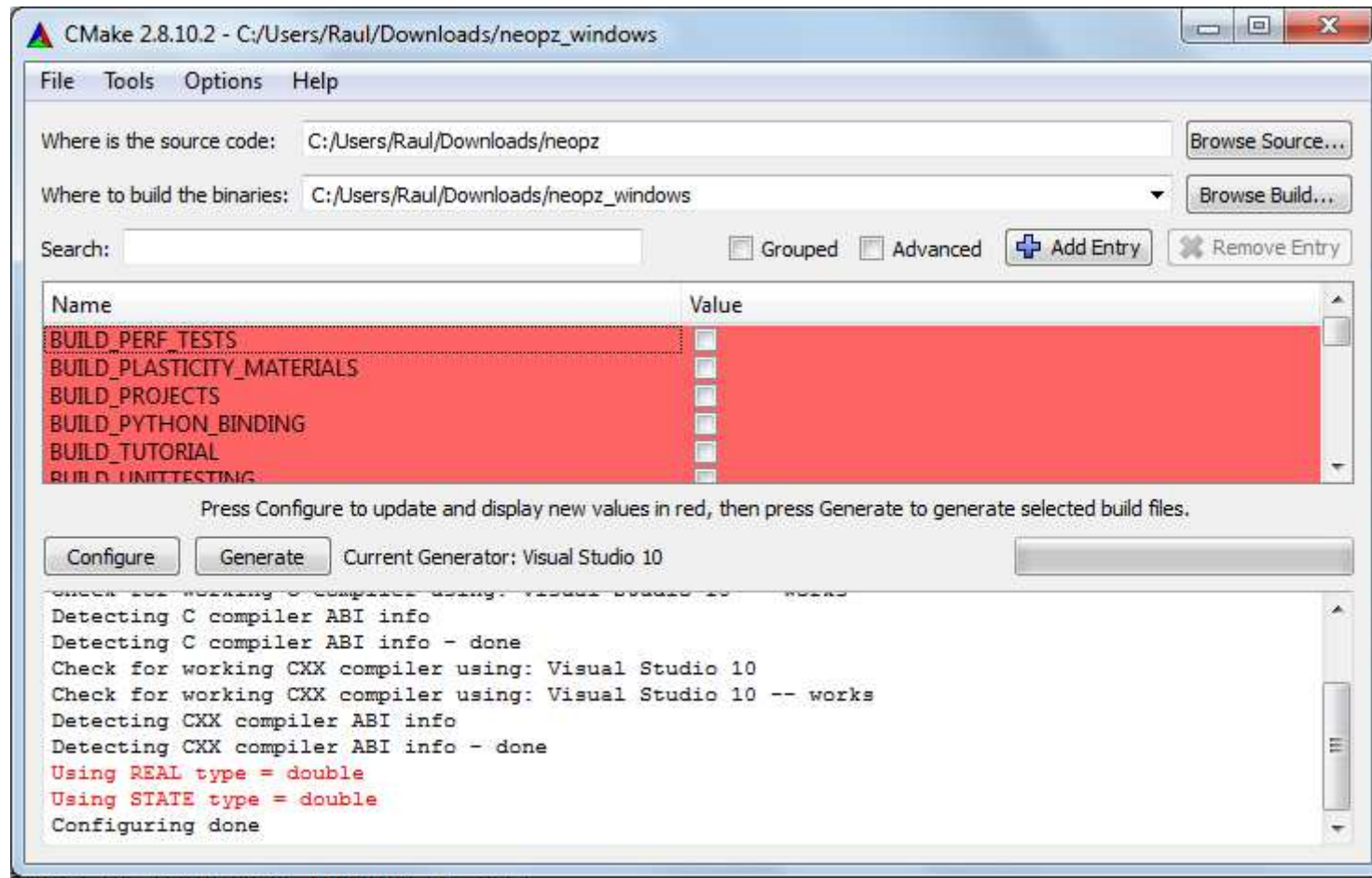
- É um pacote de ferramentas contendo:

CMake: Sistema de compilação inteligente para pequenos ou elaborados projetos (1ª etapa)

CTest, CDash: Usadas para testar o projeto e publicar seus testes na WEB (2ª etapa)

CPack: Usada para criar um pacote com instalador p/ distribuição (3ª etapa)

CMake - Interface



CMake - CDash

Login All Dashboards Wednesday, February 05 2014 16:42:04 BRST

 **PZ**

Dashboard Calendar Previous **Current** Next Project Regression

No update data as of **Thursday, November 01 2012 - 00:00 BRST** Show Filters Advanced View Auto-refresh Help

Nightly

Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
www.labmec.org.br-PZ-WIN-VS10	 PZ-Debug-All flags on	0	0	0	0	50				Nov 01, 2012 - 02:21 BRST
www.labmec.org.br-PZ-WIN-VS10	 PZ-Release-All flags on	0	0	0	0	50				Nov 01, 2012 - 02:01 BRST
www.labmec.org.br-PZ-MACOSX	 PZ-Debug-BOOST OFF ⓘ	0	0	0	0	3				Nov 01, 2012 - 02:06 BRST
www.labmec.org.br-PZ-MACOSX	 PZ-Debug-FAD OFF ⓘ	0	0	0	0	3				Nov 01, 2012 - 01:46 BRST
www.labmec.org.br-PZ-MACOSX	 PZ-Debug-LOG4CXX OFF ⓘ	0	0	0	0	1				Nov 01, 2012 - 00:15 BRST
www.labmec.org.br-PZ-MACOSX	 PZ-Debug-METIS OFF ⓘ	0	0	0	0	1	0	0	65	Nov 01, 2012 - 00:46 BRST
www.labmec.org.br-PZ	 PZ-Debug-All flags on - float	0	0	0	0	1				Nov 01, 2012 - 01:30 BRST
www.labmec.org.br-PZ	 PZ-Debug-BOOST OFF	0	0	0	0	1				Nov 01, 2012 - 02:42 BRST
www.labmec.org.br-PZ	 PZ-Debug-FAD OFF	0	0	0	0	1				Nov 01, 2012 - 03:16 BRST
www.labmec.org.br-PZ	 PZ-Debug-LOG4CXX OFF	0	0	0	0	1				Nov 01, 2012 - 03:49 BRST

CMake - Sintaxe

- É uma linguagem
- Tipo básico de dados: String
- Não diferencia maiúsculas e minúsculas
- Suporta expressões regulares
- Sintaxe básica:

```
comando( arg1 arg2 arg3 ... )
```

CMake - Sintaxe

- **Comentário:**

```
#Eu sou um comentário
```

- **Atribuição de variáveis:**

```
#var = valor1  
set (var valor1)  
#texto = "valor1 valor2"  
set (texto "valor1 valor2")  
#lista = "valor1;valor2;valor3"  
set (lista valor1 valor2 valor3)  
#copia_var = valor de var -> valor1  
set (copia_var ${var})
```

OBS: Ñ precisa declarar → criada no primeiro uso

CMake - Sintaxe

- Mensagens:

```
MESSAGE (TIPO "Mensagem a ser exibida")
```

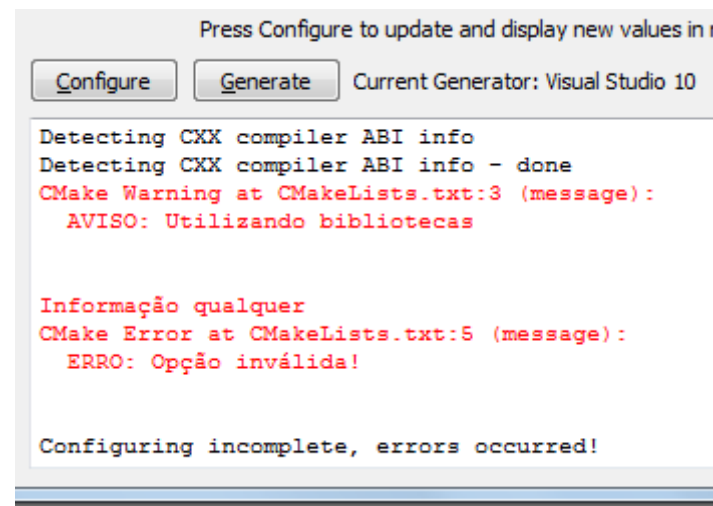
Onde TIPO:

- (sem tipo): Informação importante
- STATUS: Informação eventual
- WARNING: Aviso → continua processamento
- FATAL_ERROR: Erro → aborta processamento

CMake - Sintaxe

- Exemplo:

```
message (WARNING "AVISO: Utilizando bibliotecas  
${LIBS} ")  
message ("Informação qualquer")  
message (FATAL_ERROR "ERRO: Opção inválida!")
```



Press Configure to update and display new values in r

Current Generator: Visual Studio 10

```
Detecting CXX compiler ABI info  
Detecting CXX compiler ABI info - done  
CMake Warning at CMakeLists.txt:3 (message):  
  AVISO: Utilizando bibliotecas  
  
Informação qualquer  
CMake Error at CMakeLists.txt:5 (message):  
  ERRO: Opção inválida!  
  
Configuring incomplete, errors occurred!
```

CMake - Sintaxe

- Opções:

Mostra uma opção para o usuário selecionar:

```
OPTION (VARIAVEL_OP "Mensagem sobre a opção" ON/OFF)
```

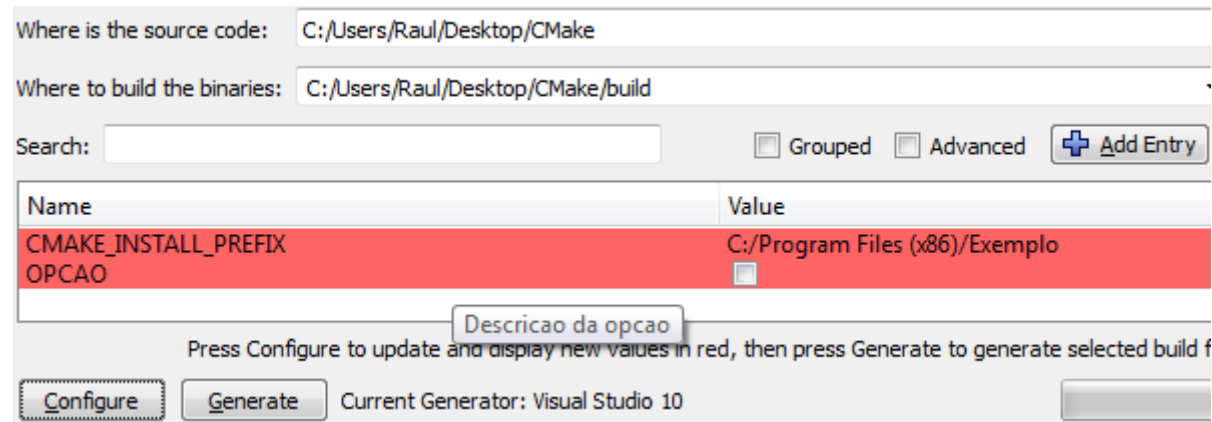
- Podem ser:

- Liga/Desliga uso de bibliotecas
- Liga/Desliga *flags* de compilação (-DFLAG)
- etc

CMake - Sintaxe

- Opções - Exemplo:

OPTION (OPCAO "Descrição da opção" OFF)



CMake - Sintaxe

- **Condicional:**

```
IF (var)  
    comando (...)  
ENDIF (var)
```

Falso: "", 0, N, NO, OFF, FALSE, NOTFOUND, ***-NOTFOUND.

Verdadeiro: 1, ON, TRUE, Y, YES, YE e tudo não listado em falso

CMake - Sintaxe

- **AND**

```
IF(variavel1 AND variavel2)
```

- **OR**

```
IF(variavel1 OR variavel2)
```

- **NOT**

```
IF(NOT variavel)
```

CMake - Sintaxe

- **LESS / GREATER / EQUAL**

```
IF(variable LESS number)
```

```
IF(variable GREATER number)
```

```
IF(variable EQUAL number)
```

CMake - Sintaxe

- **Condicional - Exemplo:**

```
SET( num 4 )  
# testa "num" é maior que 100  
IF( num GREATER 100 )  
    MESSAGE( "O numero ${number} muito  
grande." )  
ENDIF( num GREATER 100 )
```

CMake - Sintaxe

- **Laços:**

```
FOREACH (valor ${lista})  
    comando(... Usando ${valor})  
ENDFOREACH (valor)
```

```
WHILE (condição)  
    comando (...)  
ENDWHILE (condição)
```


CMake - Sintaxe

- **Laços - Exemplos:**

```
#### Exemplo usando FOREACH
SET (lista a b c)
FOREACH (valor ${lista})
    MESSAGE (${valor})
ENDFOREACH(valor)
```

```
#### Exemplo usando WHILE
SET (num 0)
WHILE( num LESS 11 )
    MESSAGE( "${num}" )
    MATH( Expr num "${num} + 1" ) # incrementa num
ENDWHILE( num LESS 11 )
```

CMake - Sintaxe

- Nomeia o projeto e, opcionalmente, defini suas linguagens:

```
PROJECT ( <NOME_PROJ> <LING1> <LING2>  
... )
```

CMake - Sintaxe

- Lista de diretórios de *include*
- Onde pré-processador procurará por arquivos .h

```
INCLUDE_DIRECTORIES ( INCLUDE_DIRS )
```

CMake - Sintaxe

- Adiciona um subdiretório à árvore de build

```
ADD_SUBDIRECTORY (SUBDIR)
```

- CMake procura por CMakeLists.txt dentro de cada subdiretório

CMake - Sintaxe

- Adicionando um executável ao projeto:

```
ADD_EXECUTABLE (<NOME_EXEC> fonte1  
fonte2 fonteN)
```

CMake - Sintaxe

- Ligando executável à bibliotecas:

```
TARGET_LINK_LIBRARIES (<NOME_EXEC>  
lib1 lib2 libN)
```

Agenda

- Controle de versão
- SVN
- SVN - Exemplo
- Processo de build
- CMake
- **CMake - Exemplo**

CMake

Exemplo simples

```
% gcc -g ex01.cpp -o ex01
```

CMakeLists.txt

```
#Especificar a versão usada para criar o projeto  
cmake_minimum_required (VERSION 2.6)  
#Especificar o nome do projeto  
project (Exemplos)  
#Adiciona a opção -g na compilação  
set (CMAKE_BUILD_TYPE "Debug")  
#Criará um executável ex01 usando o código ex01.cpp  
add_executable (ex01 ex01.cpp)
```


CMake

Exemplo com biblioteca

```
% gcc -g mult_matrix.cpp -o mult_matrix -lblas
```

CMakeLists.txt

```
#Especificar a versão usada para criar o projeto
cmake_minimum_required (VERSION 2.6)
#Especificar o nome do projeto
project (Exemplo_com_biblioteca)
#Adiciona a opção -g na compilação
set (CMAKE_BUILD_TYPE "Debug")
#Criará um executável usando o código CPP
add_executable (mult_matrix mult_matrix.cpp)
#Ligará o executável com a biblioteca BLAS
target_link_libraries (mult_matrix blas)
```

CMake

Exemplos

Exemplo mais complexo:

<http://code.google.com/p/neopz/source/browse/trunk/CMakeLists.txt>

Exemplo utilizando compilador Fortran:

<http://www.cmake.org/Wiki/CMakeForFortranExample>

Links

SVN:

<http://subversion.apache.org/>

<http://svnbook.red-bean.com/>

CMake:

<http://www.cmake.org/cmake/help/syntax.html>

<http://www.cmake.org/Wiki/CMake>