CINT C++ Interpreter
update

ROOT2001 at Fermi-Lab

Masaharu Goto

Agenda

- About CINT
- Progress since ROOT2000
- Major Progresses
- Issues
- Plans

About CINT

- CINT is a C/C++ Interpreter
- Compiled C/C++ library can be dynamically linked
- Platform independent (almost all major OS, MPU)
- Not perfect, but covers 95%/85% of C/C++ syntax
- Since 1996, CINT is used as a macro interpreter and dictionary generator in ROOT framework
- Many users world wide

Progress since ROOT2000

- Weekly release
  - Released every weekend
  - Revision 5.14.32-90 , 5.15.02

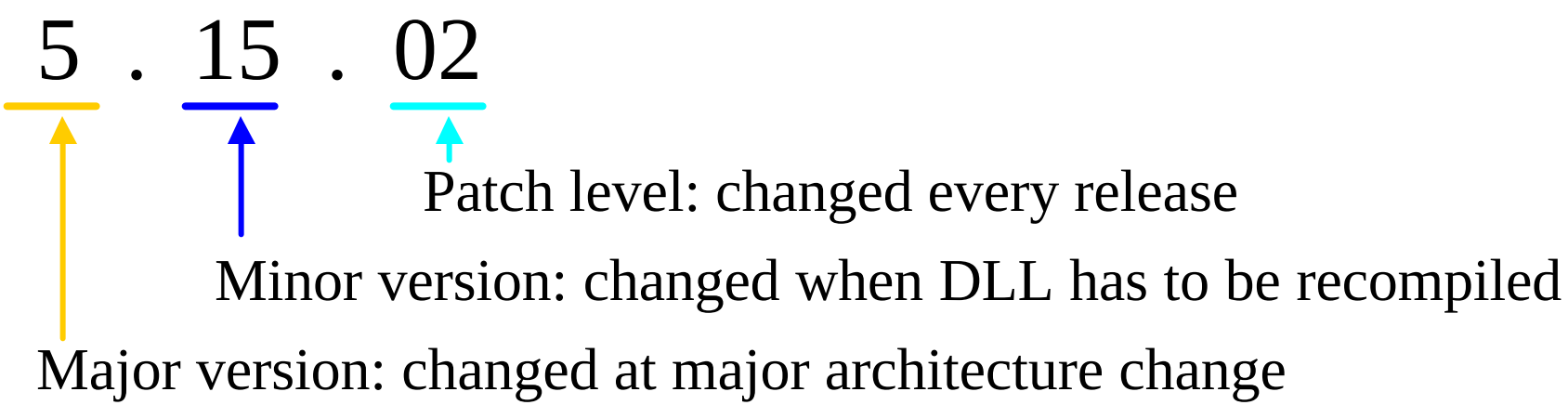
- Many bug fixes and enhancements
  - Mods serial number : 1276 to 1538
  - In case of side-effects, new features can be turned off by defining G__OLDIMPLEMENTATION[ModsNum] in G__ci.h

Major Progresses

- New revision number policy and DLL compatibility
- Multi-threading (experimental)
- New memory allocation system
- Variable arguments in interpreted code and x86 architecture
- Stub class/function

New revision number policy

- Cint revision number will be synchronized with DLL binary compatibility
- Benefit
  - Easy to figure out DLL binary compatibility
  - (Cint table size became independent from DLL binary compatibility)



New revision number policy (2)

- In most cases, DLL created by an older minor version can be loaded by newer minor version. But, the other way around is not possible.
  - DLL created by 5.15.xx could be loaded by 5.16.xx, but not always. If not possible, it is rejected by DLL revision checking mechanism.
  - DLL created by 5.16.xx will not be loaded by 5.15.xx.

Multi Threading (Experimental)

- “True” multi-thread interpretation using a very smart trick proposed by Christoph Bugel and his colleague in TTI-telecom
- Copy libcint.so/dll, load them separately and run in multiple threads
- Works best on Windows
  - RedHat6.2 Linux is also supported, but with limitation.
- Need special configuration
- Refer to $CINTSYSDIR/demo/multilibcint/README.txt

Multi Threading (2)

- Can not apply this technique on ROOT/CINT
  - Problem in Use-Model and platform dependency
  - Problem in Linux dynamic loader symbol resolution
  - ROOT static object ?
  - At this moment, other technique work better
- More investigation is needed

Issues

- Precompiled STL container in ROOT environment
  - Problem associated with memory allocation system
- Frequent bug reports on following items
  - Bytecode optimization (Try '.O0' - '.O3' if you have problem)
  - '#pragma link' statement
- RedHat7.0 Linux and Windows2000 support
  - Minor improvement may be needed
- Undetermined multi-thread execution model
  - Use-Case and OS limitation

Plans

- Precompiled STL container in ROOT environment
  - Find solution during ROOT2001
- Multi-thread investigation
- Continue support

Your Bug Report is appreciated

- Report bug reports to rootdev@pcroot.cern.ch
  - roottalk may be overlooked
- Please report CINT revision number too
  - If it is a degrade from previous version, what is the last revision that worked fine.
- Your simplified example is very helpful
  - Thank you very much