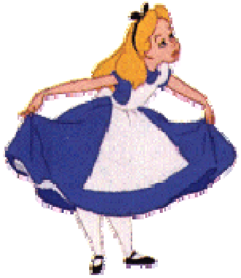
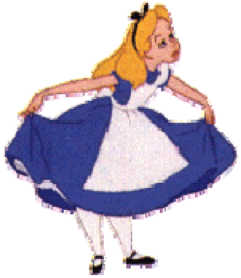


ALICE Offline Tutorial

F.Carminati, P.Christakoglou, J.F.Grosse-Oetringhaus, P.Hristov, A.Peters, P.Saiz

CERN, January 22, 2007



Part I: AliRoot

References

- AliRoot “Primer” http://aliceinfo.cern.ch/Offline/AliRoot/primer.html
- Presentation of A.Morsch during the International Workshop on Computing for Heavy Ion Physics (April 26th 2005)
- Tutorial prepared by B.Nilsen
- Presentations of Yu.Belikov and P.Christakoglou: Alice offline week (March 2006)

Outline

- AliRoot installation
  - Root
  - Geant3
  - Fluka
  - AliRoot

Outline I

- Versioning systems
  - Concurrent Versions System (CVS)
- Debugging tools
  - Compilers
  - Debuggers
  - Profilers
  - Run time and memory management tools

Outline II

- Simulation
  - Generators
  - Configuration (Config.C)
- Reconstruction
- Analysis
  - ESD classes
  - Selectors

Environment Variables

```
# ROOT
 export ROOTSYS=<prefix>/root
 export PATH=$PATH\:$ROOTSYS/bin
 export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$ROOTSYS/lib

 # AliRoot
 export ALICE=<prefix>/alice
 export ALICE_ROOT=$ALICE/AliRoot
 export ALICE_TARGET=`root-config --arch`
 export PATH=$PATH\:$ALICE_ROOT/bin/tgt_${ALICE_TARGET}
 export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$ALICE_ROOT/lib/tgt_${ALICE_TARGET}

 # GEANT 3
 export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$ALICE/geant3/lib/tgt_${ALICE_TARGET}

 # FLUKA
 export FLUPRO=<prefix>/fluka
```

For Mac OSX replace LD_LIBRARY_PATH with DYLD_LIBRARY_PATH

Installation of AliEn

- Download and execute the AliEn installer
  - wget http://alien.cern.ch/alien-installer
  - chmod a+x alien-installer
  - ./alien-installer
- Install client and gshell API

Root Download (CVS)

- Login (once) to the ROOT CVS repository
  - cvs –d :pserver:cvs@root.cern.ch:/user/cvs login
  - CVS password: cvs
- Download (check out) the needed ROOT version (v5-14-00 in the example):
  - cvs -qz9 -d :pserver:cvs@root.cern.ch:/user/cvs co -r v5-14-00 root

Root Configuration: Example script

```
#!/bin/sh

ALIEN_ROOT=<prefix>/alien

./configure \
     --with-pythia6-uscore=SINGLE \
     --enable-cern --enable-rfio \
     --enable-mathmore --enable-mathcore --enable-roofit \
     --enable-asimage --enable-minuit2 \
     --enable-alien \
     --with-alien-incdir=${ALIEN_ROOT}/api/include \
     --with-alien-libdir=${ALIEN_ROOT}/api/lib
```

Note that you MUST have installed AliEn before!

Root: Compilation, Tests

- After the configuration optionally edit $ROOTSYS/MyConfig.mk (for example add there OPT += -g)
- Do “cd $ROOTSYS; make; make map”
- Do “cd test; make”
- Add “.” (dot) to (DY)LD_LIBRARY_PATH
- Run “stress”, ”stressGeom”, ”stressLinear”, “stressVector”, etc.
- Now you have fully operational Root ☺

Installation of Geant3

- Download (must be in $ALICE/geant3!)
  - cd $ALICE
  - cvs -qz9 -d :pserver:cvs@root.cern.ch:/user/cvs co -r v1-6 geant3
  - cd geant3
  - make
- Expect some warnings!
- Needs root to be installed before!

Fluka

- Register and get the Fluka library from http://www.fluka.org
- Unpack the library in $ALICE/fluka
  - cd $ALICE/fluka
  - tar xvfz ~/fluka....tgz
  - ln -s neuxsc_72.bin neuxsc.bin

AliRoot

- Login (once) to the AliRoot CVS repository
  - cvs –d :pserver:cvs@alisoft.cern.ch:/soft/cvsroot login
  - CVS password: cvs
- cd $ALICE
- Download (check out) the needed AliRoot version (v4-04-Release in the example)
  - cvs -qz9 -d :pserver:cvs@alisoft.cern.ch:/soft/cvsroot co -r v4-04-Release AliRoot
- cd AliRoot; make
- Expect warnings!

.rootrc

- Defines ROOT/AliRoot default setting
- cp $ALICE_ROOT/.rootrc ~/
- vi ~/.rootrc
- Add to “Unix.*.Root.MacroPath:” “$ALICE_ROOT/macros” and any other directory where you want to use macros from
- Add “Unix.*.Root.IncludePath:” with directories where include files you want are kept

AliRoot make options

- make [module]
  - If no module given, check dependencies and compile everything that might have changed
  - Check dependencies in given module and then compile everything that has changed there
  - Non default targets: htmldoc, alilibs, aliroot, makedistr, profile, depend, TFluka
- Cleaning up

clean-all	clean up everything including cern libraries
clean-module	ITS,TPC,... just that subdirectory
clean-modules	clean all modules (not cern libraries)
clean-libs	clean all libraries (not object files)
clean-aliroot	clean up all aliroot libraries
distclean	clean as if fresh install

- make -k clean-XXXX is suggested

AliRoot update & make “quirks”

- Update your distribution to the latest patches of a release
  - cd $ALICE_ROOT; cvs -qz9 up -Pd; make
- Update your distribution to a given tag
  - cd $ALICE_ROOT; cvs -qz9 up -Pd -r <revision tag>; make
- Sometimes (deleted files) it fails with the message (e.g.)
  - “No rule to make XXXX needed by YYYY”
  - Just clean and remake
  - make -k □clean-<the culprit module>; make
- If everything else fails
  - make -k clean-aliroot ; make
- If you are getting desperate
  - make -k clean-all ; make

QuickTime™ and a
TIFF (Uncompressed) decompress
are needed to see this picture.

Concurrent Versions System (CVS)

- http://www.cvshome.org/
- CVS facilitates parallel/concurrent code development
- Easy support and simple access
- Possibility to establish group permissions
  - e.g. only detector experts and CVS administrators can commit code to given detector module
- Rich set of commands
- A lot of visualization/logging/control tools

Common CVS Commands

- export CVSROOT=:pserver:cvs@alisoft.cern.ch:/soft/cvsroot
  - Only for fist c[heck]o[ut]
- login stores password in .cvspass
- checkout retrieves the source files
  - cvs -qz9 co -r v4-04-Release AliRoot
- update retrieves modifications from the HEAD in the repository and merges them to the local ones
  - cvs -qz9 up -AdP STEER
- diff shows differences between the local and repository versions
  - cvs -qz9 diff STEER

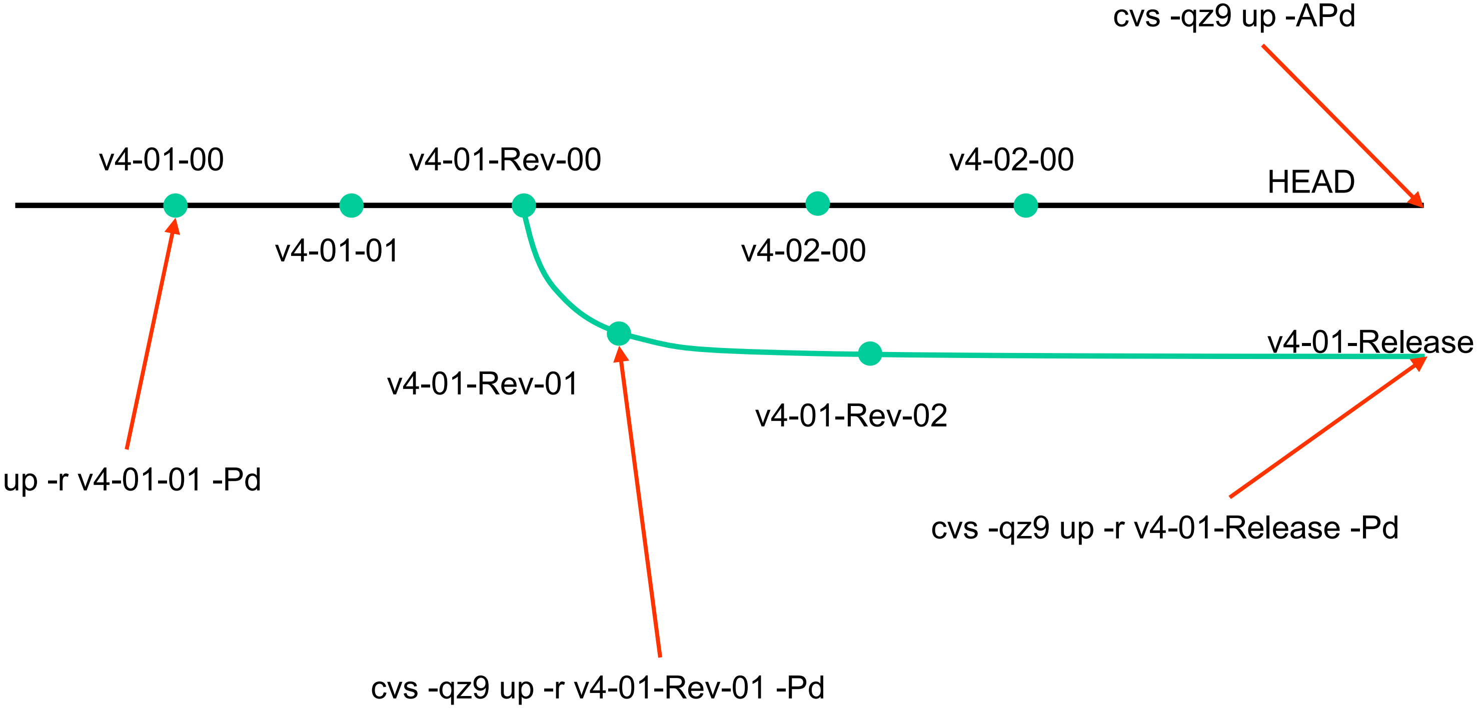
Common CVS Commands

- add adds files or directories to the repository
  - cvs -qz9 add AliTPCseed.*
  - You still have to commit to have the files actually added
- remove (rm) removes old files or directories from the repository
  - cvs -qz9 remove -f CASTOR
- commit (ci) checks in the local modifications to the repository and increments the version
  - cvs -qz9 ci -m “Coding convention” STEER
  - Please use meaningful comments!
- log finds the story of mods for a file
  - cvs -qz9 log STEER/AliRun.h

Main CVS Commands

- tag creates new tags and/or branches
  - cvs -qz9 tag -b v4-04-Release   # this creates a branch
  - cvs -qz9 tag v4-04-Rev-11   # this creates a tag
  - Tags are cheap!!
- status returns the actual status of a file: revision, sticky tag, dates, options, and local modifications
  - cvs -qz9 status [-v] Makefile
- logout removes the stored password
  - Hardly ever used

cvs structure for AliRoot



cvs -qz9 up -APd

v4-01-00

v4-01-Rev-00

v4-02-00

HEAD

v4-01-01

v4-02-00

v4-01-Release

v4-01-Rev-01

v4-01-Rev-02

cvs -qz9 up -r v4-01-01 -Pd

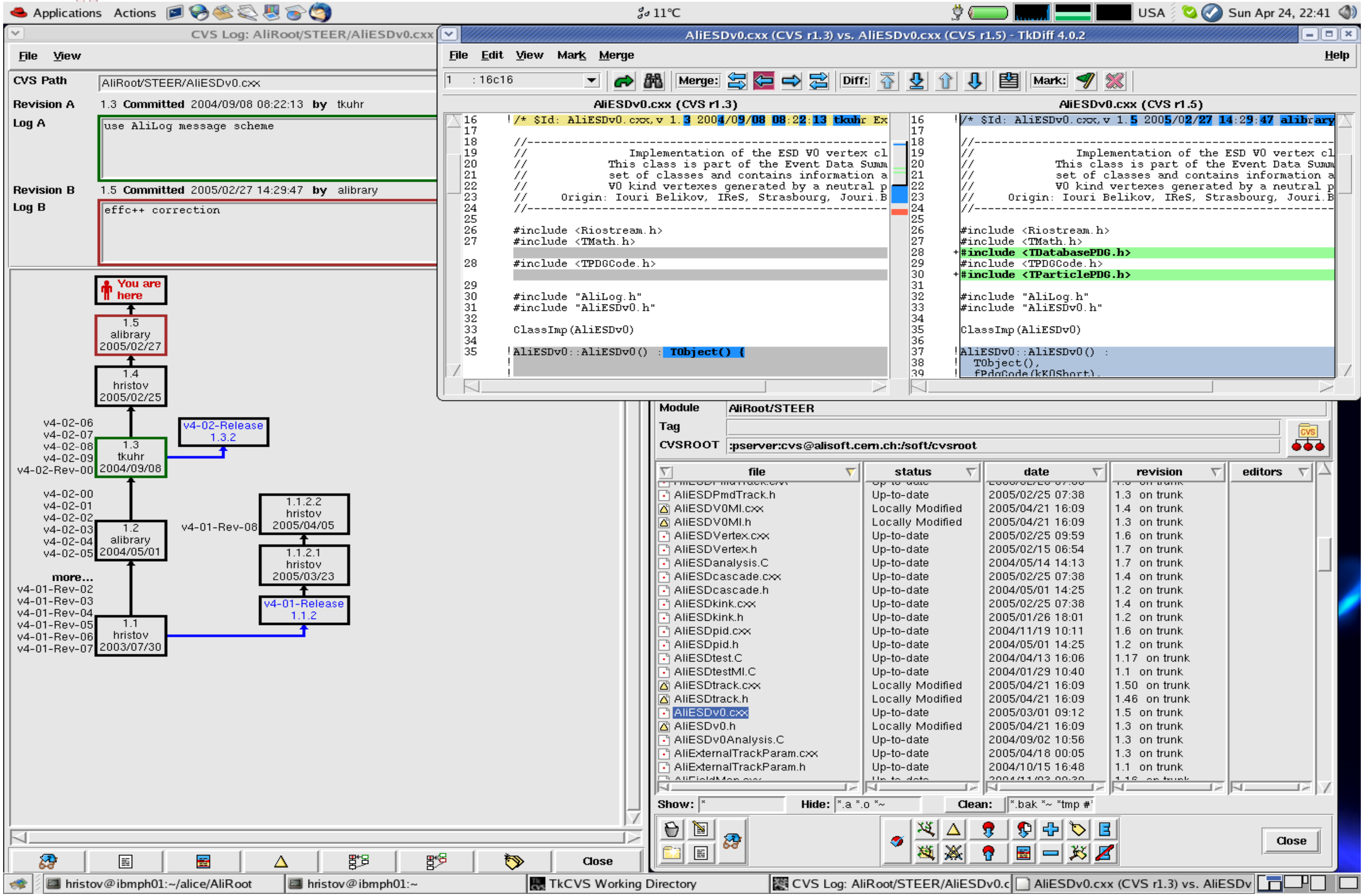
cvs -qz9 up -r v4-01-Release -Pd

cvs -qz9 up -r v4-01-Rev-01 -Pd

CVS Visualization/Logging/Control Tools

- TkCVS
  - http://www.twobarleycorns.net/tkcvs.html
- CVSWeb
  - http://www.freebsd.org/projects/cvsweb.html
- Cervisia: KDE distribution
  - http://www.kde.org/apps/cervisia/
- cvs2cl.pl: Producing ChangeLog
  - http://www.red-bean.com/cvs2cl/

TkCVS Tool



CVS Log: AliRoot/STEER/AliESDv0.cxx

CVS Path: AliRoot/STEER/AliESDv0.cxx

Revision A: 1.3 **Committed** 2004/09/08 08:22:13 **by** tkuhr

Log A:

```
use AliLog message scheme
```

Revision B: 1.5 **Committed** 2005/02/27 14:29:47 **by** alibrary

Log B:

```
effc++ correction
```

AliESDv0.cxx (CVS r1.3) vs. AliESDv0.cxx (CVS r1.5) - TkDiff 4.0.2

AliESDv0.cxx (CVS r1.3)

```
!/* $Id: AliESDv0.cxx,v 1.3 2004/09/08 08:22:13 tkuhr Ex

 //-------------------------------------------------------
 //                 Implementation of the ESD V0 vertex cl
 //              This class is part of the Event Data Summ
 //              set of classes and contains information a
 //              V0 kind vertexes generated by a neutral p
 //      Origin: Iouri Belikov, IReS, Strasbourg, Jouri.B
 //-------------------------------------------------------

 #include <Riostream.h>
 #include <TMath.h>

 #include <TPDGCode.h>


 #include "AliLog.h"
 #include "AliESDv0.h"

 ClassImp(AliESDv0)

!AliESDv0::AliESDv0() : TObject() {
!
!
```

AliESDv0.cxx (CVS r1.5)

```
!/* $Id: AliESDv0.cxx,v 1.5 2005/02/27 14:29:47 alibrary

 //-------------------------------------------------------
 //                 Implementation of the ESD V0 vertex cl
 //              This class is part of the Event Data Summ
 //              set of classes and contains information a
 //              V0 kind vertexes generated by a neutral p
 //      Origin: Iouri Belikov, IReS, Strasbourg, Jouri.B
 //-------------------------------------------------------

 #include <Riostream.h>
 #include <TMath.h>
+#include <TDatabasePDG.h>
 #include <TPDGCode.h>
+#include <TParticlePDG.h>

 #include "AliLog.h"
 #include "AliESDv0.h"

 ClassImp(AliESDv0)

!AliESDv0::AliESDv0() :
!  TObject(),
!  fPdgCode(kK0Short),
```

Module: AliRoot/STEER

Tag:

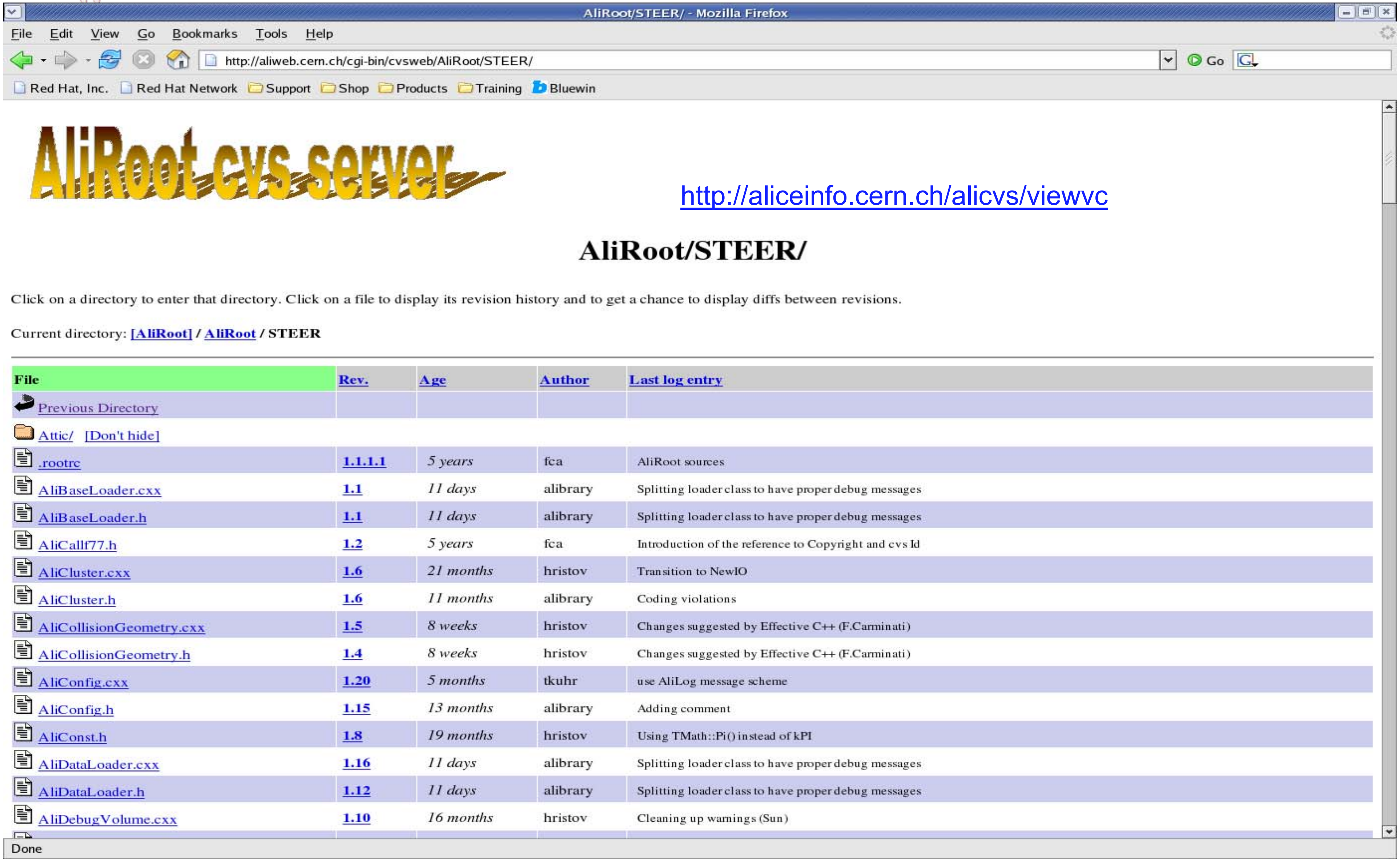
CVSROOT: :pserver:cvs@alisoft.cern.ch:/soft/cvsroot

file	status	date	revision	editors
AliESDPmdTrack.h	Up-to-date	2005/02/25 07:38	1.3 on trunk	
AliESDV0MI.cxx	Locally Modified	2005/04/21 16:09	1.4 on trunk	
AliESDV0MI.h	Locally Modified	2005/04/21 16:09	1.3 on trunk	
AliESDVertex.cxx	Up-to-date	2005/02/25 09:59	1.6 on trunk	
AliESDVertex.h	Up-to-date	2005/02/15 06:54	1.7 on trunk	
AliESDanalysis.C	Up-to-date	2004/05/14 14:13	1.7 on trunk	
AliESDcascade.cxx	Up-to-date	2005/02/25 07:38	1.4 on trunk	
AliESDcascade.h	Up-to-date	2004/05/01 14:25	1.2 on trunk	
AliESDkink.cxx	Up-to-date	2005/02/25 07:38	1.4 on trunk	
AliESDkink.h	Up-to-date	2005/01/26 18:01	1.2 on trunk	
AliESDpid.cxx	Up-to-date	2004/11/19 10:11	1.6 on trunk	
AliESDpid.h	Up-to-date	2004/05/01 14:25	1.2 on trunk	
AliESDtest.C	Up-to-date	2004/04/13 16:06	1.17 on trunk	
AliESDtestMI.C	Up-to-date	2004/01/29 10:40	1.1 on trunk	
AliESDtrack.cxx	Locally Modified	2005/04/21 16:09	1.50 on trunk	
AliESDtrack.h	Locally Modified	2005/04/21 16:09	1.46 on trunk	
AliESDv0.cxx	Up-to-date	2005/03/01 09:12	1.5 on trunk	
AliESDv0.h	Locally Modified	2005/04/21 16:09	1.3 on trunk	
AliESDv0Analysis.C	Up-to-date	2004/09/02 10:56	1.3 on trunk	
AliExternalTrackParam.cxx	Up-to-date	2005/04/18 00:05	1.3 on trunk	
AliExternalTrackParam.h	Up-to-date	2004/10/15 16:48	1.1 on trunk	

Show: * Hide: *.a *.o *~ Clean: *.bak *~ *tmp #*

Close
hristov@ibmph01:~/alice/AliRoot
hristov@ibmph01:~
TkCVS Working Directory
CVS Log: AliRoot/STEER/AliESDv0.c
AliESDv0.cxx (CVS r1.3) vs. AliESDv

CVSWeb



AliRoot/STEER/ - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://aliweb.cern.ch/cgi-bin/cvsweb/AliRoot/STEER/
Go
Red Hat, Inc. Red Hat Network Support Shop Products Training Bluewin
AliRoot cvs server
Done

http://aliceinfo.cern.ch/alicvs/viewvc

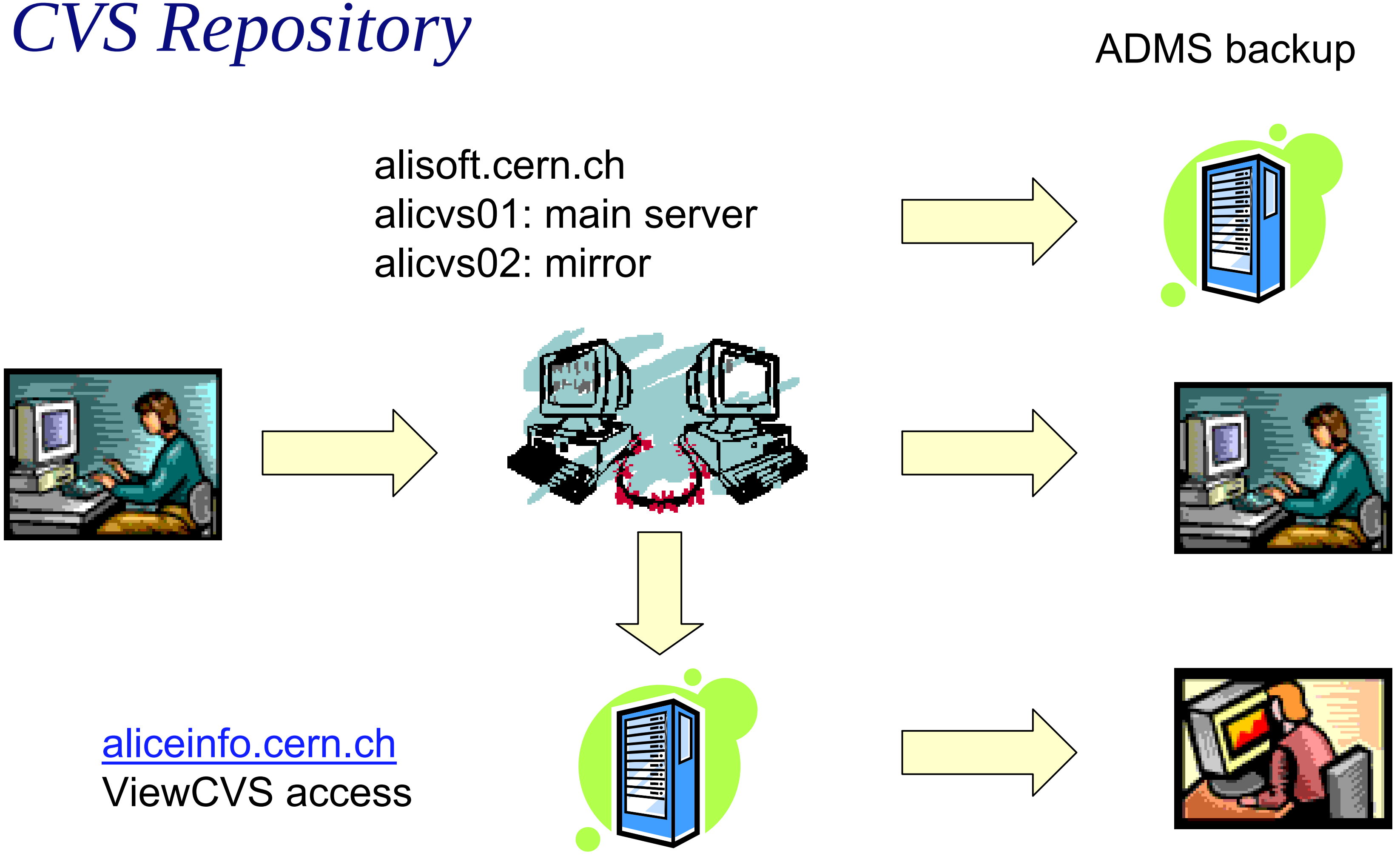
AliRoot/STEER/

Click on a directory to enter that directory. Click on a file to display its revision history and to get a chance to display diffs between revisions.

Current directory: [AliRoot] / AliRoot / STEER

File	Rev.	Age	Author	Last log entry
Previous Directory				
Attic/ [Don't hide]				
.rootrc	1.1.1.1	*5 years*	fca	AliRoot sources
AliBaseLoader.cxx	1.1	*11 days*	alibrary	Splitting loader class to have proper debug messages
AliBaseLoader.h	1.1	*11 days*	alibrary	Splitting loader class to have proper debug messages
AliCallf77.h	1.2	*5 years*	fca	Introduction of the reference to Copyright and cvs Id
AliCluster.cxx	1.6	*21 months*	hristov	Transition to NewIO
AliCluster.h	1.6	*11 months*	alibrary	Coding violations
AliCollisionGeometry.cxx	1.5	*8 weeks*	hristov	Changes suggested by Effective C++ (F.Carminati)
AliCollisionGeometry.h	1.4	*8 weeks*	hristov	Changes suggested by Effective C++ (F.Carminati)
AliConfig.cxx	1.20	*5 months*	tkuhr	use AliLog message scheme
AliConfig.h	1.15	*13 months*	alibrary	Adding comment
AliConst.h	1.8	*19 months*	hristov	Using TMath::Pi() instead of kPI
AliDataLoader.cxx	1.16	*11 days*	alibrary	Splitting loader class to have proper debug messages
AliDataLoader.h	1.12	*11 days*	alibrary	Splitting loader class to have proper debug messages
AliDebugVolume.cxx	1.10	*16 months*	hristov	Cleaning up warnings (Sun)

CVS Repository



ADMS backup

alisoft.cern.ch
alicvs01: main server
alicvs02: mirror

aliceinfo.cern.ch
ViewCVS access

Code Maintenance

- Supported platforms:
  - Linux (Pentium, Itanium, and Opteron with gcc and icc compilers)
  - Sun: SPARC, x86 (Solaris with CC compiler)
  - Alpha (was OSF with cxx compiler, now Linux with gcc)
  - Mac/ppc and Mac/Intel (Darwin with gcc and icc compilers)

Compilers

- Linux:
  - gcc (versions 3.2 - 3.4.6). GNU license, free source distribution. Not very strict or ANSI compliant by default. Works also with 4.0.x
  - icc (Intel) versions 7.0 – 9.0. Free for non-commercial use, high performance compiler. Gives 20-30% improvement during the execution of AliRoot/Root. More difficult to debug.
- Sun CC: extensive warnings, capable to detect non-initialized variables, etc. Integrated in Forte with debugger, profiler, and memory checker.
- Alpha cxx: strict compiler (relatively old), finds come problems not indicated by any other compiler
- Mac gcc 4.0.x and icc: FORTRAN now OK, using g95, but gfortran is also getting there

Debugging

- Errors during the execution
  - Floating point exceptions
    - Illegal operations: division by zero, sqrt of negative argument, assignment of NaN, etc.
  - Segmentation violations/faults
    - Attempt to access a forbidden memory location
  - Bus error
    - Attempt to access memory that the computer cannot address

Debugging

- Using printf(...), cout and assert(...)
  - Often the only easy way to find the problem
  - assert(...) aborts the program execution if the argument is FALSE. Is is a macro from assert.h, you can remove its action by compiling with -DNDEBUG
- Using gdb
  - Needs compilation with -g -O0 option
    - Optimisation prevents proper debugging
  - One can use it directly (gdb aliroot) or attach it to a process (gdb aliroot 12345)
  - FORTRAN support is very bad

Main GDB Commands

- run starts the execution of the program
- where prints the program stack
- quit exits the gdb session
- break sets break point
  - break AliLoader.cxx:100
  - break 'AliLoader::AliLoader()’
- delete removes break point
- cont continues the run
- watch sets watchpoint: watch *fData (slow!)
- list shows the source code
- print evaluates and prints expression
- help for the rest of the commands...

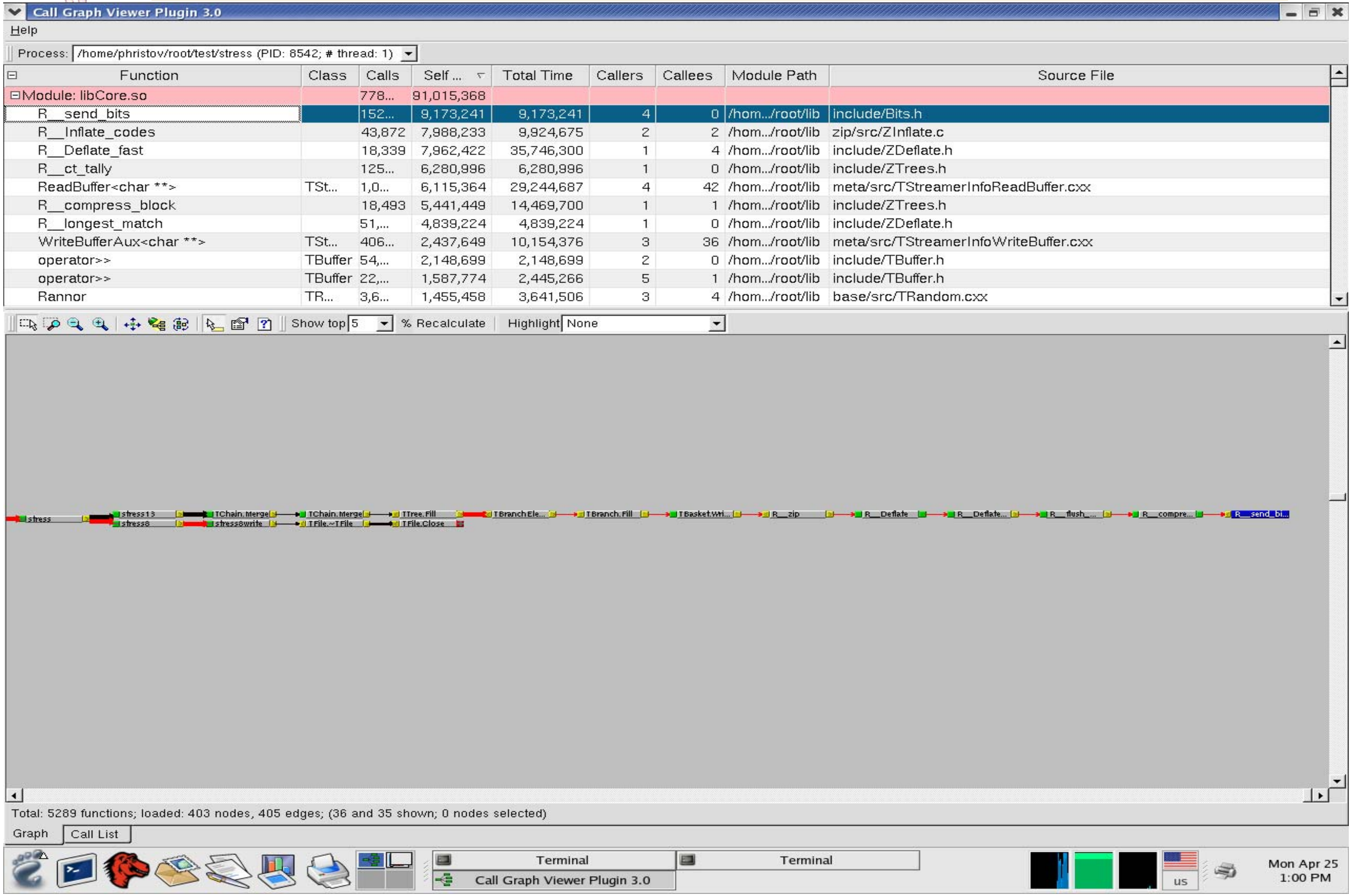
Profiling

- Linux
  - gprof: compilation with -pg option, static libraries
  - Oprofile: uses kernel module
  - VTune: instruments shared libraries
- Sun
  - Sun workshop (Forte agent): needs compilation with profiling option (-pg)
- Alpha
  - Pixie profiler: instruments shared libraries for profiling

VTune Profiling Tool

- Available from Intel Web site, free for non-commercial use on Linux
  - Unavailable elsewhere
- Possibility for call-graph and sampling profiling
- Instruments shared libraries, needs only -g option
- Example session
  - vtl activity stress -c callgraph -app $ROOTSYS/test/stress,” -b” -moi $ROOTSYS/test/stress
  - vtl run stress
  - vtl show
  - vtl view stress::r1 -gui

VTune



Process: /home/phristov/root/test/stress (PID: 8542; # thread: 1)

Function	Class	Calls	Self ...	Total Time	Callers	Callees	Module Path	Source File
Module: libCore.so		778...	91,015,368					
R__send_bits		152...	9,173,241	9,173,241	4	0	/hom.../root/lib	include/Bits.h
R__Inflate_codes		43,872	7,988,233	9,924,675	2	2	/hom.../root/lib	zip/src/ZInflate.c
R__Deflate_fast		18,339	7,962,422	35,746,300	1	4	/hom.../root/lib	include/ZDeflate.h
R__ct_tally		125...	6,280,996	6,280,996	1	0	/hom.../root/lib	include/ZTrees.h
ReadBuffer<char **>	TSt...	1,0...	6,115,364	29,244,687	4	42	/hom.../root/lib	meta/src/TStreamerInfoReadBuffer.cxx
R__compress_block		18,493	5,441,449	14,469,700	1	1	/hom.../root/lib	include/ZTrees.h
R__longest_match		51,...	4,839,224	4,839,224	1	0	/hom.../root/lib	include/ZDeflate.h
WriteBufferAux<char **>	TSt...	406...	2,437,649	10,154,376	3	36	/hom.../root/lib	meta/src/TStreamerInfoWriteBuffer.cxx
operator>>	TBuffer	54,...	2,148,699	2,148,699	2	0	/hom.../root/lib	include/TBuffer.h
operator>>	TBuffer	22,...	1,587,774	2,445,266	5	1	/hom.../root/lib	include/TBuffer.h
Rannor	TR...	3,6...	1,455,458	3,641,506	3	4	/hom.../root/lib	base/src/TRandom.cxx

Total: 5289 functions; loaded: 403 nodes, 405 edges; (36 and 35 shown; 0 nodes selected)

Valgrind: detection of run time errors

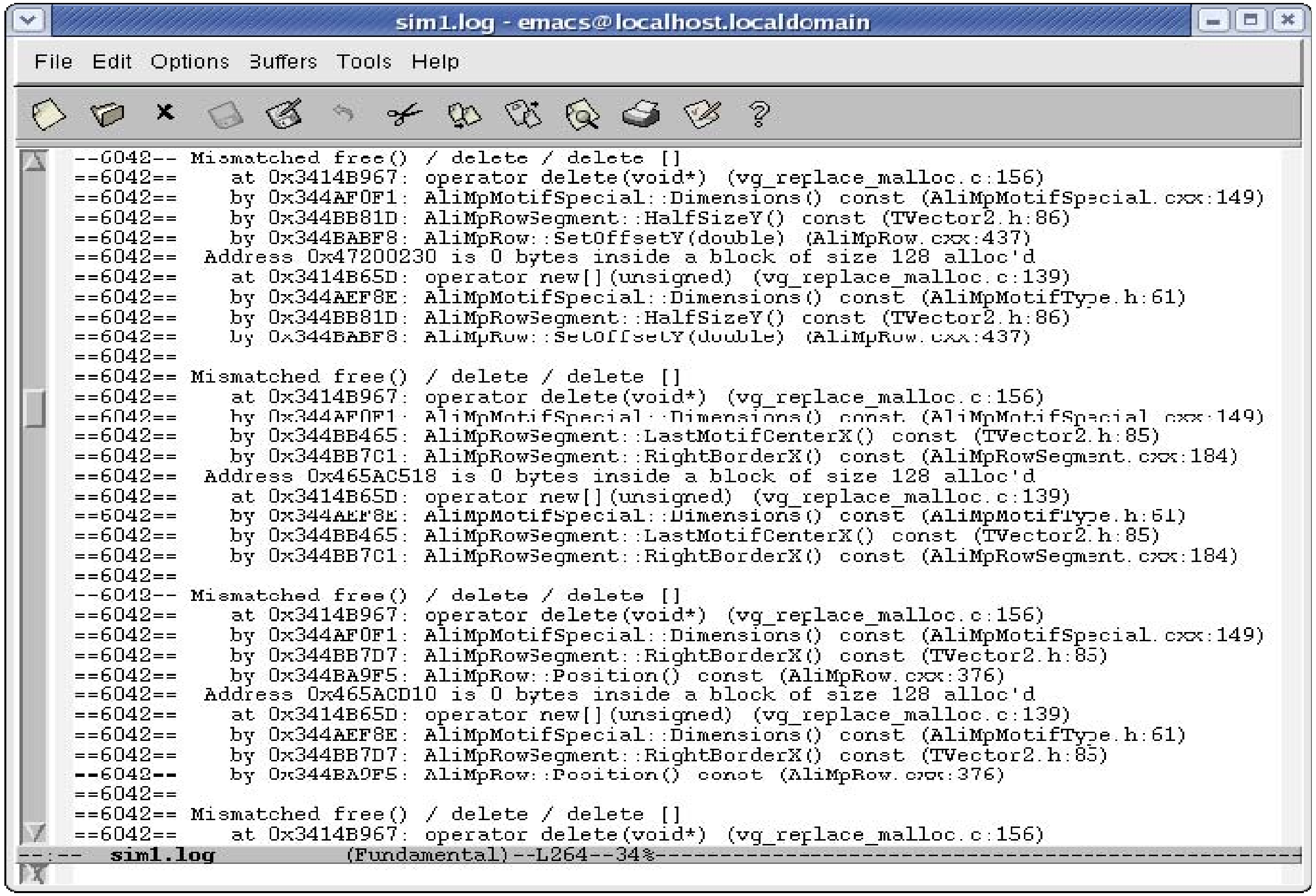
- http://www.valgrind.org/
- Set of tools
  - Memcheck for memory management problems
  - Addrcheck: lightweight memory checker
  - Cachegrind: cache profiler
  - Massif: heap profiler
  - Hellgrind: thread debugger
  - Callgrind: extended version of Cachegrind

Valgrind: Memcheck

- Memcheck can detect:
  - Use of non-initialised memory
  - Reading/writing memory after it has been free'd
  - Reading/writing off the end of malloc'd blocks
  - Reading/writing inappropriate areas on the stack
  - Memory leaks - where pointers to malloc'd blocks are lost forever
  - Mismatched use of malloc/new/new [] vs free/delete/delete []
  - Overlapping src and dst pointers in memcpy() and related functions
  - Some misuses of the POSIX pthreads API

Valgrind Example

- valgrind --tool=addrcheck --error-limit=no aliroot -b -q sim.C


sim1.log - emacs@localhost.localdomain
File Edit Options Buffers Tools Help

```
==6042== Mismatched free() / delete / delete []
==6042==    at 0x3414B967: operator delete(void*) (vg_replace_malloc.c:156)
==6042==    by 0x344AF0F1: AliMpMotifSpecial::Dimensions() const (AliMpMotifSpecial.cxx:149)
==6042==    by 0x344BB81D: AliMpRowSegment::HalfSizeY() const (TVector2.h:86)
==6042==    by 0x344BABF8: AliMpRow::SetOffsetY(double) (AliMpRow.cxx:437)
==6042==  Address 0x47200230 is 0 bytes inside a block of size 128 alloc'd
==6042==    at 0x3414B65D: operator new[](unsigned) (vg_replace_malloc.c:139)
==6042==    by 0x344AEF8E: AliMpMotifSpecial::Dimensions() const (AliMpMotifType.h:61)
==6042==    by 0x344BB81D: AliMpRowSegment::HalfSizeY() const (TVector2.h:86)
==6042==    by 0x344BABF8: AliMpRow::SetOffsetY(double) (AliMpRow.cxx:437)
==6042==
==6042== Mismatched free() / delete / delete []
==6042==    at 0x3414B967: operator delete(void*) (vg_replace_malloc.c:156)
==6042==    by 0x344AF0F1: AliMpMotifSpecial::Dimensions() const (AliMpMotifSpecial.cxx:149)
==6042==    by 0x344BB465: AliMpRowSegment::LastMotifCenterX() const (TVector2.h:85)
==6042==    by 0x344BB7C1: AliMpRowSegment::RightBorderX() const (AliMpRowSegment.cxx:184)
==6042==  Address 0x465AC518 is 0 bytes inside a block of size 128 alloc'd
==6042==    at 0x3414B65D: operator new[](unsigned) (vg_replace_malloc.c:139)
==6042==    by 0x344AEF8E: AliMpMotifSpecial::Dimensions() const (AliMpMotifType.h:61)
==6042==    by 0x344BB465: AliMpRowSegment::LastMotifCenterX() const (TVector2.h:85)
==6042==    by 0x344BB7C1: AliMpRowSegment::RightBorderX() const (AliMpRowSegment.cxx:184)
==6042==
==6042== Mismatched free() / delete / delete []
==6042==    at 0x3414B967: operator delete(void*) (vg_replace_malloc.c:156)
==6042==    by 0x344AF0F1: AliMpMotifSpecial::Dimensions() const (AliMpMotifSpecial.cxx:149)
==6042==    by 0x344BB7D7: AliMpRowSegment::RightBorderX() const (TVector2.h:85)
==6042==    by 0x344BA9F5: AliMpRow::Position() const (AliMpRow.cxx:376)
==6042==  Address 0x465ACD10 is 0 bytes inside a block of size 128 alloc'd
==6042==    at 0x3414B65D: operator new[](unsigned) (vg_replace_malloc.c:139)
==6042==    by 0x344AEF8E: AliMpMotifSpecial::Dimensions() const (AliMpMotifType.h:61)
==6042==    by 0x344BB7D7: AliMpRowSegment::RightBorderX() const (TVector2.h:85)
==6042==    by 0x344BA9F5: AliMpRow::Position() const (AliMpRow.cxx:376)
==6042==
==6042== Mismatched free() / delete / delete []
==6042==    at 0x3414B967: operator delete(void*) (vg_replace_malloc.c:156)
--:--  sim1.log          (Fundamental)--L264--34%-------------------------------------------
```

Root Memory Checker

- Detection of memory leaks, statistics of memory usage
- Fast, easy to use
  - Re-link aliroot with -lNew
    - Modify the $ALICE_ROOT/Makefile
    - rm $ALICE_ROOT/tgt_${ALICE_TARGET}/aliroot
    - make
  - Add ”Root.MemCheck: 1” in .rootrc
  - Run the program: aliroot -b -q sim.C
  - Run “memprobe -e aliroot”
  - Check *.info files
- Does not work with the latest version of root – being fixed

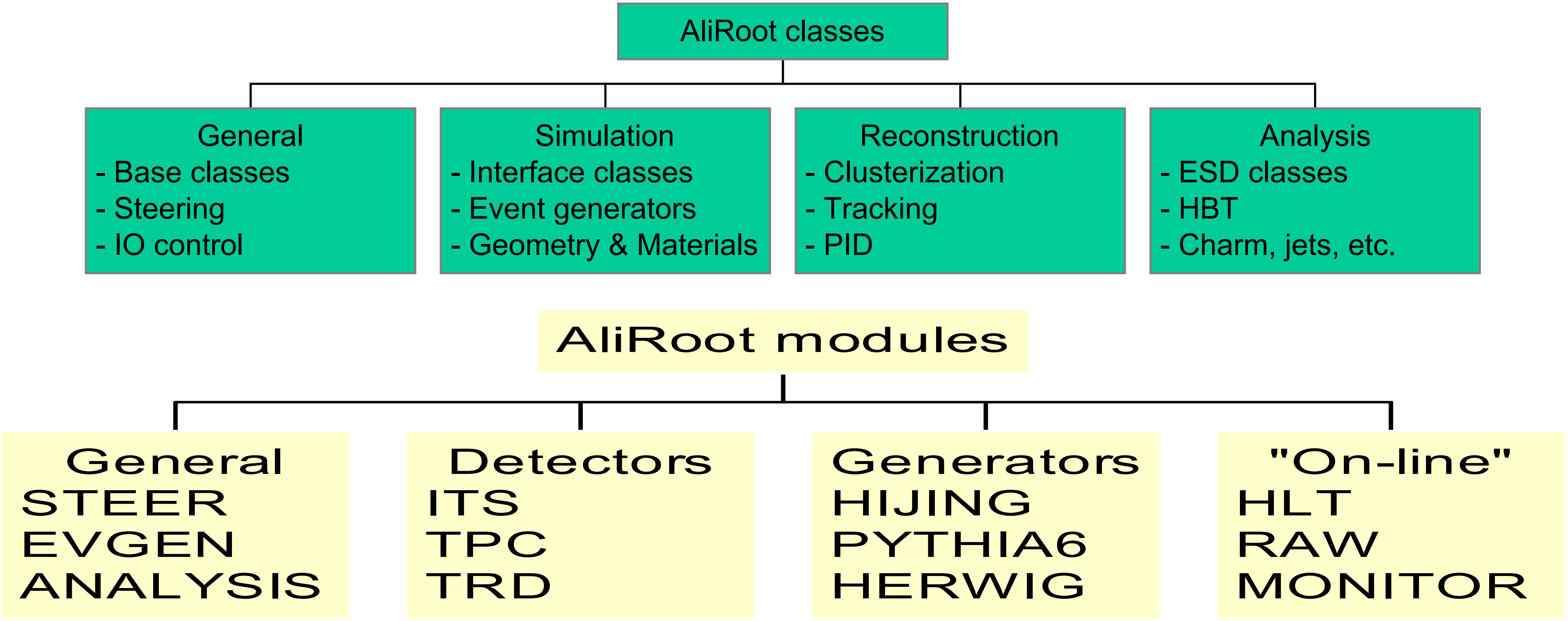
Code Checking and Reverse Engineering Tool

- General description: http://aliceinfo.cern.ch/Offline/AliRoot/Coding-Conventions.html
- Installation: unpack the tarball from /afs/cern.ch/alice/library/local/IRSTCodeAnalysisTool.tgz, in the directory $ALICE/local/IRST
- Do “make check-all” in $ALICE_ROOT
- Do “make reveng-all” in $ALICE_ROOT
- Do “make revdisp-all” in $ALICE_ROOT

Release Policy

- The code compiles on all the platforms.
- Extensive tests on Linux (Pentium, Itanium, Opteron; gcc and icc)
- Root memory checker: no significant memory leaks
- Tag (no branch): monthly
- Valgrind: fixes for run-time errors
- Profiling: gprof and Vtune
  - Fix algorithms using abnormal quantities of memory or CPU
- Tag with branch => release every 6 months

AliRoot: General Layout



AliRoot classes

General
- Base classes
- Steering
- IO control

Simulation
- Interface classes
- Event generators
- Geometry & Materials

Reconstruction
- Clusterization
- Tracking
- PID

Analysis
- ESD classes
- HBT
- Charm, jets, etc.

AliRoot modules

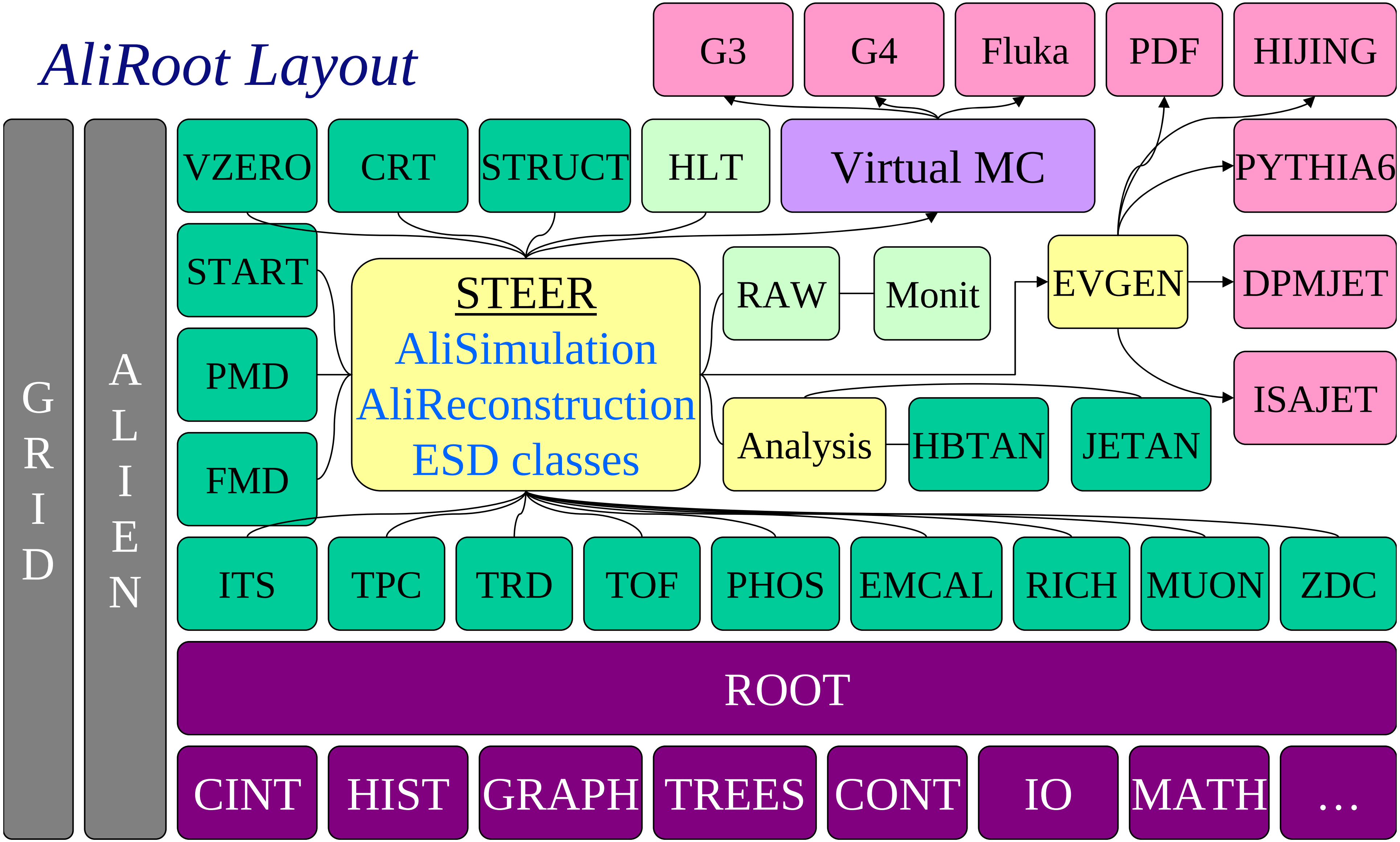
General
STEER
EVGEN
ANALYSIS

Detectors
ITS
TPC
TRD

Generators
HIJING
PYTHIA6
HERWIG

"On-line"
HLT
RAW
MONITOR

AliRoot Layout



G3
G4
Fluka
PDF
HIJING

VZERO
CRT
STRUCT
HLT
Virtual MC
PYTHIA6

START
PMD
FMD

STEER
AliSimulation
AliReconstruction
ESD classes

RAW
Monit
EVGEN
DPMJET

Analysis
HBTAN
JETAN
ISAJET

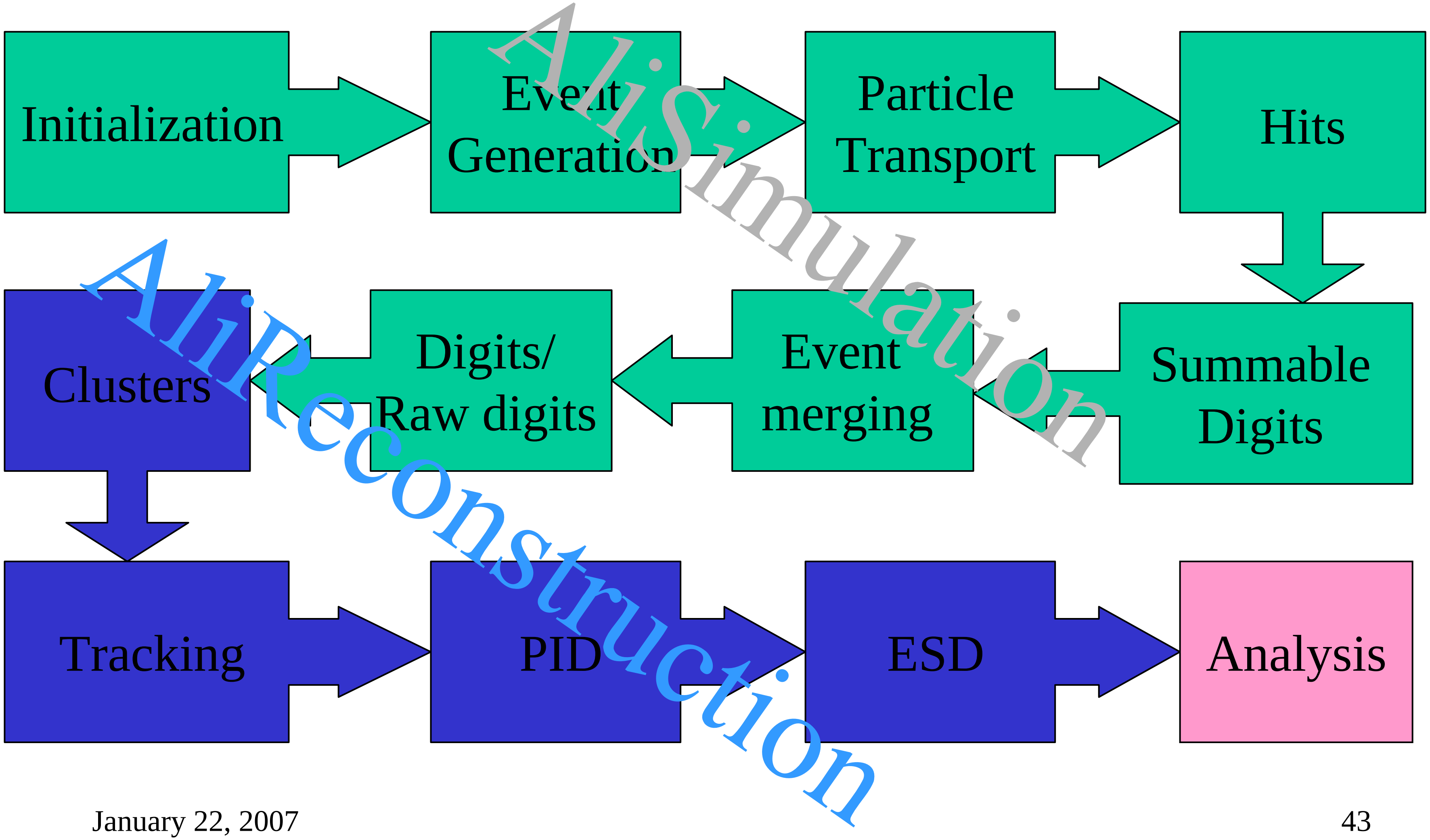
ITS
TPC
TRD
TOF
PHOS
EMCAL
RICH
MUON
ZDC

ROOT

CINT
HIST
GRAPH
TREES
CONT
IO
MATH
…

GRID
ALIEN

AliRoot: Execution Flow



Config.C: Steering the Simulation

- Sets random seed
- Creates transporter
- Creates RunLoader
- Replaces MC particle decay model by Pythia6
- Set up transporter
- Creates and sets up event simulator
- Defines ALICE Magnetic Field
- Defines All materials and geometries/detectors

```
Void Config(){
gRandom->SetSeed(123456789);
new TGeant3TGeo(“Transporter”);
AliRunLoader *rl = AliRunLoader::Open(
      “galice.root”,defaultFileNames,”recreate”);
gAlice->SetRunLoader(rl);
TVirtualMCDecayer *dec = AliDecayerPythia();
dec->Init();
gMC->SetExternalDecayer(dec);
…
gMC->SetCut(“CUTGAM”,1.e-3);
…
AliGenHIJINGpara *gen = new AliGenHIJINGpara(100);
gen->Init(); // Registers its self to gAlice
gAlice_>SetField(new AliMagFMaps(…);
AliBody *BODY = new AliBODY(“BODY”,”Alice envelope”);
// Registers itself to gAlice
```

External Generators: HIJING

- HIJING
  - HIJING (Heavy Ion Jet INteraction Generator) combines
    - A QCD-inspired model of jet production with the Lund model for jet fragmentation
    - Hard or semi-hard parton scatterings with transverse momenta of a few GeV are expected to dominate high energy heavy ion collisions
    - The HIJING model has been developed with special emphasis on the role of mini jets in pp, pA and AA reactions at collider energies

HIJING

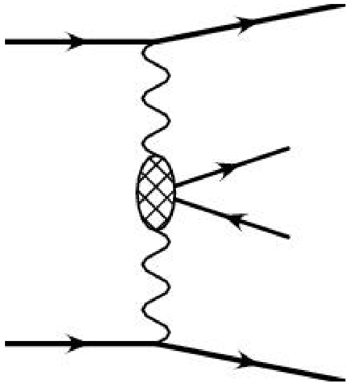
- Hijing used as
  - Underlying event
    - Realistic fluctuations (N,E) from mini-jets
    - Pessimistic multiplicity (dN/dy ~ 6000)
  - Particle Correlation studies
    - Inclusive
    - And in reconstructed jets
  - Nuclear effects
    - Shadowing
    - Quenching (parton energy loss)

Other External Generators

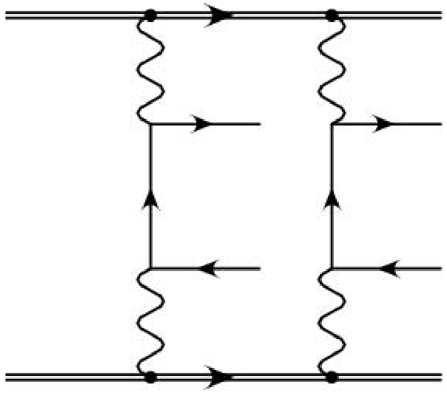
- DPMJET
  - DPMJET is an implementation of the two-component Dual Parton Model for the description of interactions involving nuclei based on the Glauber-Gribov approach
  - DPMJET treats soft and hard scattering processes in an unified way
  - The fragmentation of parton configurations is treated by the Lund model PYTHIA
- SFM (String Fusion Model)
  - The soft interactions are described by the Gribov-Regge theory of multipomeron exchange
  - The hard part of the interaction is simulated by PYTHIA and the strings formed by gluon splitting are fragmented with JETSET
  - Fusion of soft strings is included

Ultra-peripheral Collisions

AA → AA γγ → AA X



AA → AA e^+e^-



- K. Hencken et al.
- TPHIC
  - Massive particle production described in Equivalent Photon Approximation
- TEPEM
  - Electron positron pair production in UPC

pp

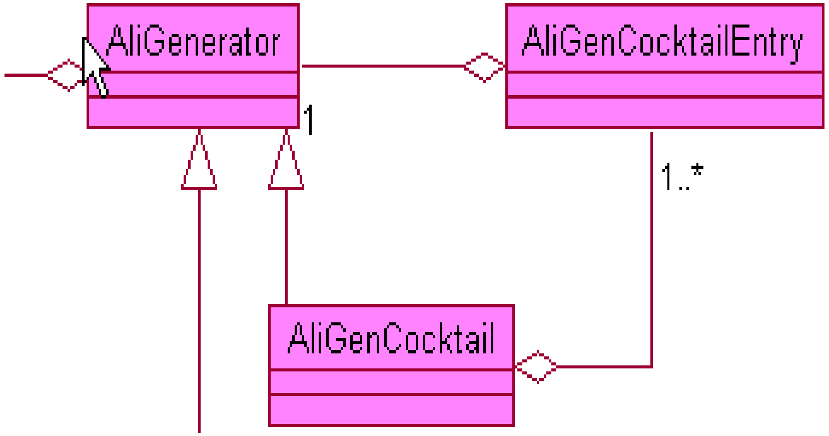
- Minimum Bias
  - Pythia, Herwig, ISAJET
  - Pythia with ATLAS Tuning
- Hard Probes
  - Pythia tuned to NLO (MNR)
    - NLO topology
  - Modification of nuclear structure functions via EKS in PDFlib

PYTHIA preconfigured processes

- Heavy Flavors (open)
  - kPyCharm, kPyBeauty
  - kPyCharmUnforced, kPyBeautyUnforced
- kPyCharmPbPbMNR, kPyD0PbPbMNR, kPyDPlusPbPbMNR, kPyBeautyPbPbMNR, kPyCharmpPbMNR, kPyD0pPbMNR, kPyDPluspPbMNR, kPyBeautypPbMNR, kPyCharmppMNR, kPyD0ppMNR, kPyDPlusppMNR, kPyBeautyppMNR
- Heavy Flavor (resonances)
  - kPyJpsi, kPyJpsiChi
- Minimum Bias
  - kPyMb, kPyMbNonDiffr
- Jets and high-pT gammas
  - kPyJets, kPyDirectGamma,
- W
  - kPyW

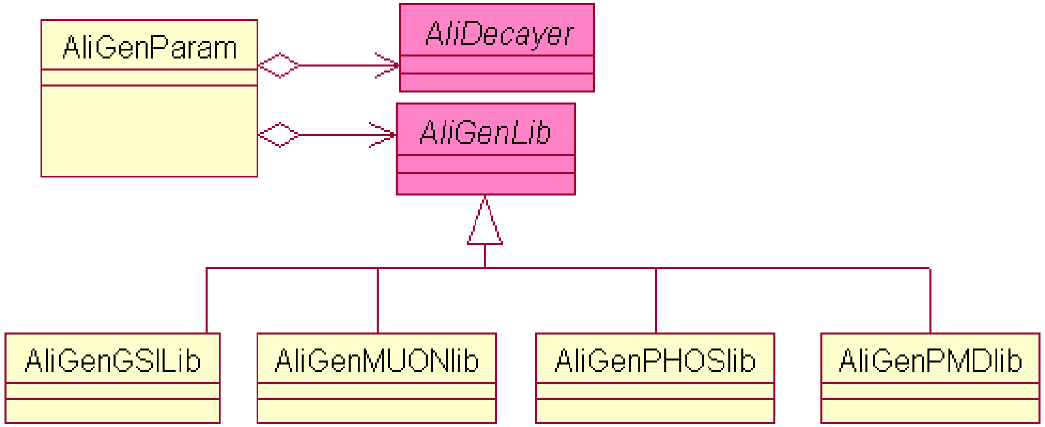
Event Generator Interfaces

- Cocktail class to assemble events, for example:
  - Underlying event + hard process
  - Different muon sources
  - pA + slow nucleons

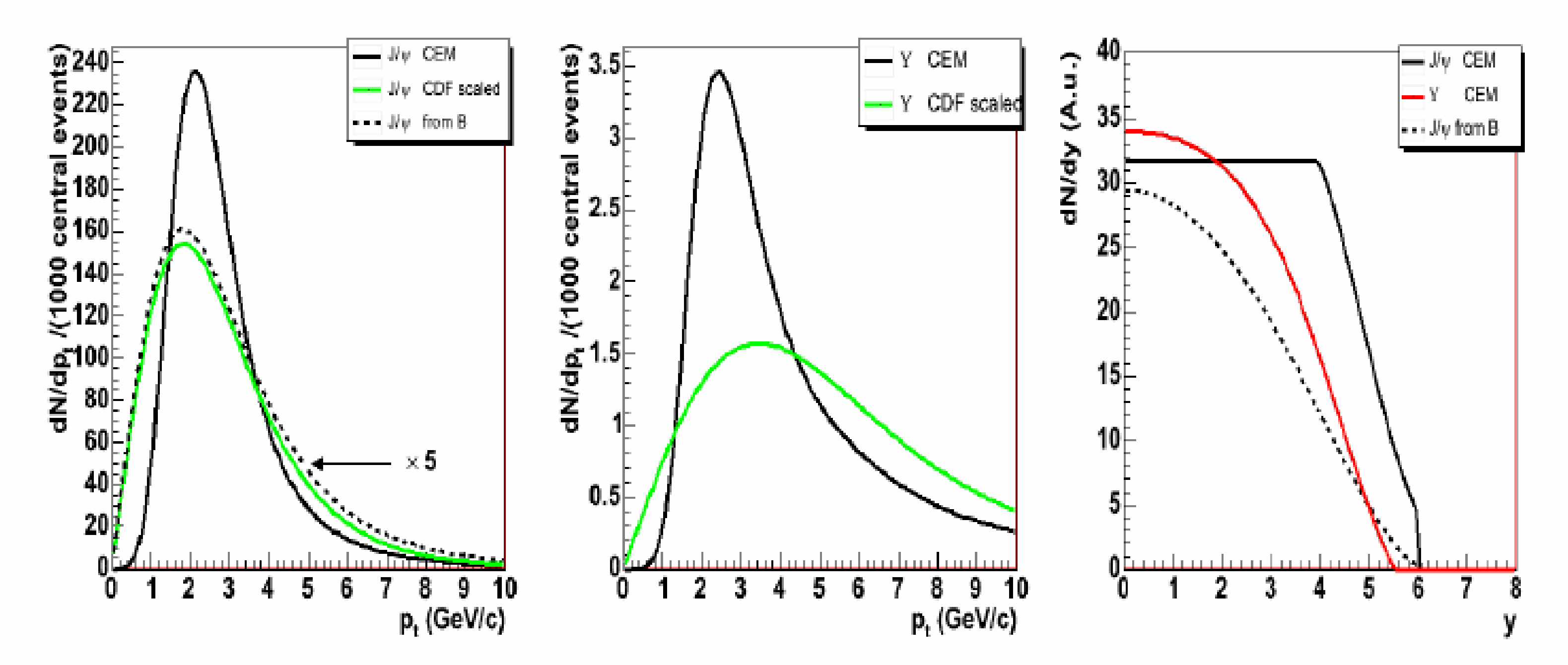


Event Generator Interfaces: Parameterizations

```
// The cocktail generator
AliGenCocktail *gener = new AliGenCocktail();

// Phi meson (10 particles)
AliGenParam *phi = new AliGenParam(10,new AliGenMUONlib(),AliGenMUONlib::kPhi,"Vogt PbPb");
phi−>SetPtRange(0, 100);
phi->SetYRange(1., +1.);
phi−>SetForceDecay(kDiElectron);

// Omega meson (10 particles)
AliGenParam *omega = new AliGenParam(10,new AliGenMUONlib(),AliGenMUONlib::kOmega,"Vogt PbPb");
omega−>SetPtRange(0, 100);
omega−>SetYRange(−1., +1.);
omega->SetForceDecay(kDiElectron);

// Adding all the components of the cocktail
gener −>AddGenerator(phi,"Phi",1);
gener −>AddGenerator(omega,"Omega",1);

// Settings, common for all components
gener −>SetOrigin(0, 0, 0);              // vertex position
gener −>SetSigma(0, 0, 5.3);             // Sigma in (X,Y,Z) (cm) on IP position
gener −>SetCutVertexZ(1.);               // Truncate at 1 sigma
gener −>SetVertexSmear(kPerEvent);
gener −>SetTrackingFlag(1);
gener >Init();
```



Example: MUON Library



Parameterisations:

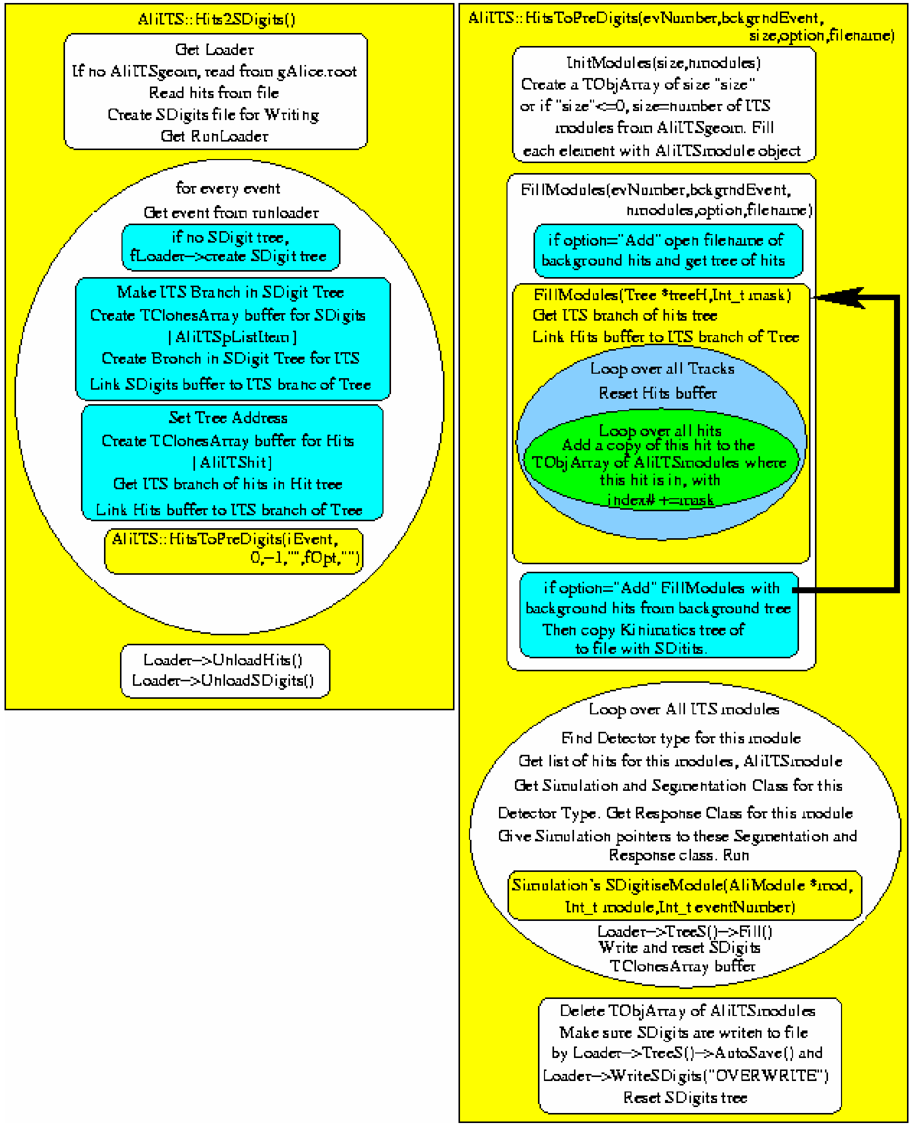
kPhi, kOmega, kEta,
kJpsi, kJpsiFamily, kPsiP, kJpsiFromB,
kUpsilon, kUpsilonFamily, kUpsilonPP,
kCharm, kBeauty,
kPion, kKaon

Run MC: Particle Transport

- Particles are transported though geometry
- At each step, a call to StepManager of the class whose volume the particle is in
- Example: AliITSvPPRasymmFMD::StepManager
  - If not sensitive volume return
  - If not charged return
  - Record Hit, particle Position (start & end of this step), Momentum, Energy lost during last step, Sub-detector in, Time of Flight, Status, Charge, and Track Number
  - In the ITS, hits can also be “merged”
- Hits are typically are deleted after SDigitization

Simulation: Summable Digits

- Apply all detector response simulation which allows results to be “merged”
  - Do not add noise
  - Do not convert AtD
  - Do not apply thresholds
- Some detectors use hits as SDigits
  - For PHOS, EMCAL the hits are already summed
  - RICH saves rings/photons



AliITS::Hits2SDigits()

Get Loader
If no AliITSgeom, read from gAlice.root
Read hits from file
Create SDigits file for Writing
Get RunLoader

for every event
Get event from runloader

if no SDigit tree,
fLoader->create SDigit tree

Make ITS Branch in SDigit Tree
Create TClonesArray buffer for SDigits
| AliITSpListItem]
Create Bronch in SDigit Tree for ITS
Link SDigits buffer to ITS branc of Tree

Set Tree Address
Create TClonesArray buffer for Hits
| AliITShit]
Get ITS branch of hits in Hit tree
Link Hits buffer to ITS branch of Tree

AliITS::HitsToPreDigits(iEvent,
0,-1,"",fOpt,"")

Loader->UnloadHits()
Loader->UnloadSDigits()

AliITS::HitsToPreDigits(evNumber,bckgrndEvent,
size,option,filename)

InitModules(size,nmodules)
Create a TObjArray of size "size"
or if "size"<=0, size=number of ITS
modules from AliITSgeom. Fill
each element with AliITSmodule object

FillModules(evNumber,bckgrndEvent,
nmodules,option,filename)

if option="Add" open filename of
background hits and get tree of hits

FillModules(Tree *treeH,Int_t mask)
Get ITS branch of hits tree
Link Hits buffer to ITS branch of Tree

Loop over all Tracks
Reset Hits buffer

Loop over all hits
Add a copy of this hit to the
TObjArray of AliITSmodules where
this hit is in, with
index# +=mask

if option="Add" FillModules with
background hits from background tree
Then copy Kinimatics tree of
to file with SDitits.

Loop over All ITS modules

Find Detector type for this module
Get list of hits for this modules, AliITSmodule
Get Simulation and Segmentation Class for this
Detector Type. Get Response Class for this module
Give Simulation pointers to these Segmentation and
Response class. Run

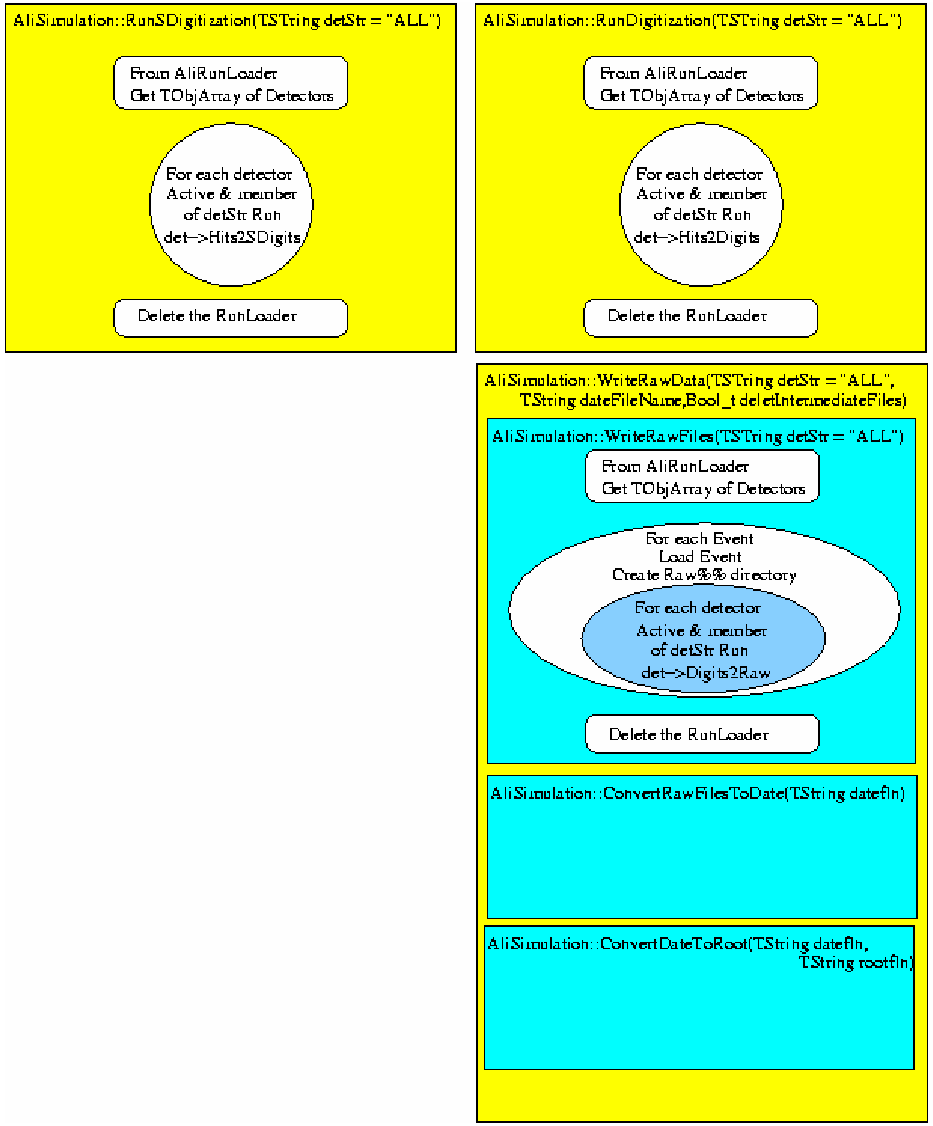
Simulation's SDigitiseModule(AliModule *mod,
Int_t module,Int_t eventNumber)

Loader->TreeS()->Fill()
Write and reset SDigits
TClonesArray buffer

Delete TObjArray of AliITSmodules
Make sure SDigits are writen to file
by Loader->TreeS()->AutoSave() and
Loader->WriteSDigits("OVERWRITE")
Reset SDigits tree

Digitization

- Adds noise
  - Random for SPD, SSD
  - Correlated for SDD
- Applies threshold
  - Simple threshold for SPD,SSD
  - 2 level threshold for SDD
- Applies ADC-ing
  - 10 bit for SDD, SSD
  - 10⇒8 conversion for SDD
- Zero suppression
  - 2 integer coordinates, 1 integer signal
  - Simulation + info by detector type



AliSimulation::RunSDigitization(TSTring detStr = "ALL")

From AliRunLoader
Get TObjArray of Detectors

For each detector
Active & member
of detStr Run
det->Hits2SDigits

Delete the RunLoader

AliSimulation::RunDigitization(TSTring detStr = "ALL")

From AliRunLoader
Get TObjArray of Detectors

For each detector
Active & member
of detStr Run
det->Hits2Digits

Delete the RunLoader

AliSimulation::WriteRawData(TSTring detStr = "ALL",
TString dateFileName,Bool_t deletIntermediateFiles)

AliSimulation::WriteRawFiles(TSTring detStr = "ALL")

From AliRunLoader
Get TObjArray of Detectors

For each Event
Load Event
Create Raw%% directory

For each detector
Active & member
of detStr Run
det->Digits2Raw

Delete the RunLoader

AliSimulation::ConvertRawFilesToDate(TString datefln)

AliSimulation::ConvertDateToRoot(TString datefln,
TString rootfln)

Reconstruction

- Possible inputs
  - DATE DDL files (only for test)
  - RAW DATE file (only for test)
  - RAW rootified file (standard format)
  - MC/Digit files (standard for simulated data)
- Local/Detector reconstruction (Files <DET>.RecPoint.root)
  - Calibration
  - Clusterisation
  - Cluster splitting...
- Vertex finder (Fills ESD)
  - Primary vertex (Z coordinate) found in SPD, and/or T0.
- Tracking (HLT and/or Barrel), filling of ESD
  - Gives final vertex from tracks and secondary vertecies.
  - HLT uses Conformal mapping (or similar) or a fast Kalman
  - Final tracking is a full Kalman
    - {TPC→ITS→TPC} →{TRD→TOF →{EMCAL|HMPID|PHOS}→TOF→TRD→TPC→ITS)
    - MUON.
- Combined PID (Fills ESD)

ESD classes, ESD shared library

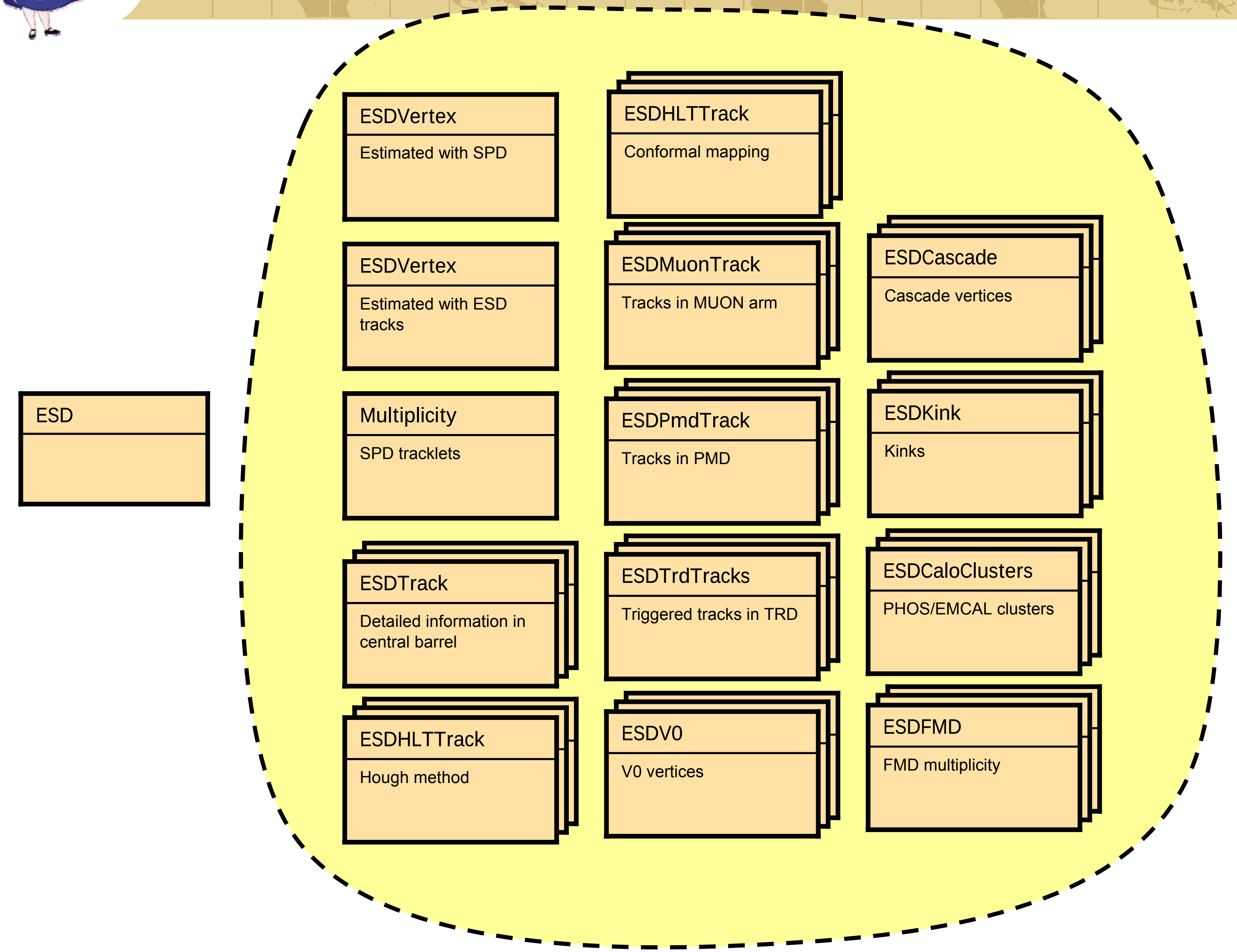
- AliESD
- AliExternalTrackParam AliESDtrack
- AliESDMuonTrack AliESDPmdTrack AliESDHLTtrack
- AliESDVertex
- AliESDv0 AliESDV0MI AliESDcascade AliESDkink
- AliESDpid AliPID
- AliKalmanTrack AliHelix AliESDV0MIParams AliTracker AliCluster AliTrackPointArray (reconstruction classes)
- AliRunTag AliLHCTag AliDetectorTag AliEventTag AliTagCreator AliTagAnalysis AliEventTagCuts

  AliXMLCollection
- AliLog

Ideally:

```
user@host> root.exe
…………………………
 root[0] gSystem->Load(“libESD”)
 root[1] .x AnyAnalysisMacro.C
```

The ESD



ESD

ESDVertex
Estimated with SPD

ESDVertex
Estimated with ESD tracks

Multiplicity
SPD tracklets

ESDTrack
Detailed information in central barrel

ESDHLTTrack
Hough method

ESDHLTTrack
Conformal mapping

ESDMuonTrack
Tracks in MUON arm

ESDPmdTrack
Tracks in PMD

ESDTrdTracks
Triggered tracks in TRD

ESDV0
V0 vertices

ESDCascade
Cascade vertices

ESDKink
Kinks

ESDCaloClusters
PHOS/EMCAL clusters

ESDFMD
FMD multiplicity

AliESD and AliESDtrack classes

- Accumulation and exchange of tracking information among the barrel detectors
- Contained in the ESD and used for physical analysis

Class AliESDtrack : public AliExternalTrackParam

- final params
- reconstruction status flags
- length, time, combined PID
- vertex constrained params
- impact parameters & cov.matrix
- params at the outer TPC wall
- params at the inner TPC wall
- …
- detector specific info (chi2, num.of clusters, PID…)

ESD Example: Loop on the tracks

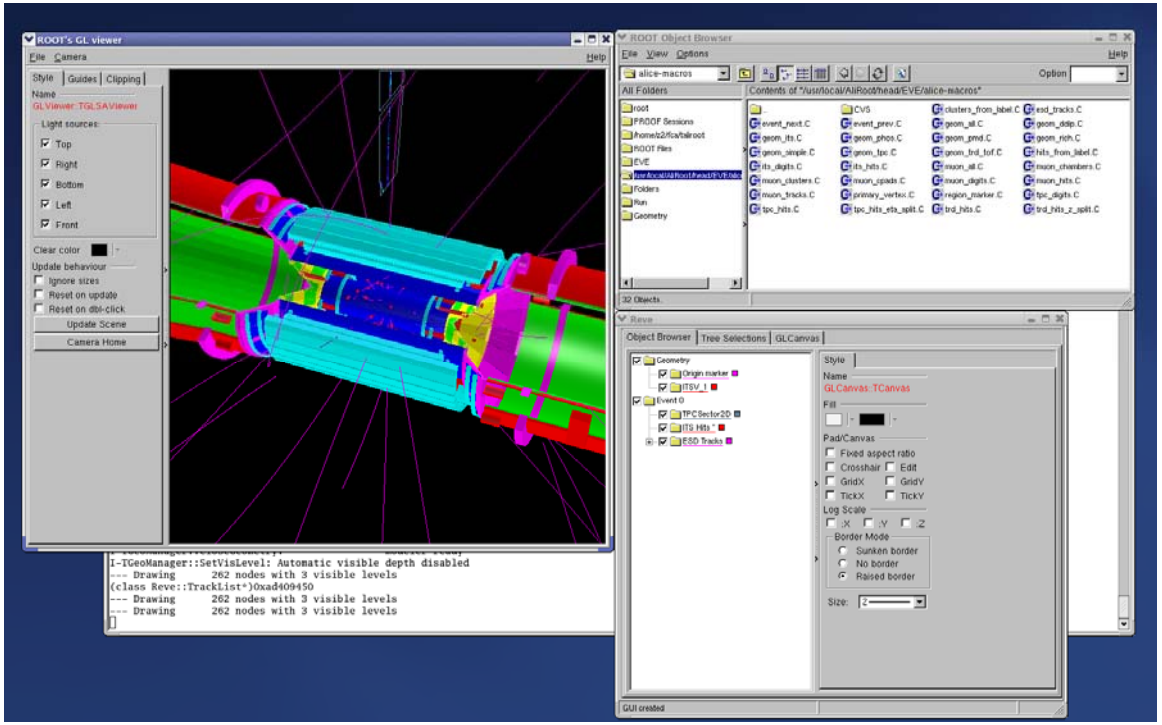
```
AliESD *event=…;                                                    //The reconstructed events are
TTree *esdTree = …;                                                 //stored in TTrees  (and so can be “chained”)

Int_t i=0;
while (esdTree->GetEvent(i++)) {                                    //loop over the reconstructed events
    …                                                       //select run, event number etc…
  if (event->GetTrigger() != … )  continue;                         //select the trigger
  AliESDvertex *primary=event->GetVertex();
  if (/* some cuts on the primary vertex */)  continue;

  Int_t ntracks=event->GetNumberOfTracks();
  for (i=0; i<ntracks; i++) {                                       //loop over ESD tracks (or kinks, V0s …)
    AliESDtrack *track=event->GetTrack(i);
    if (track->GetStatus()==…)                                      //select tracks with the proper status
    if (/* any other selection (quality) criteria  */) {
        …                                                           //do whatever with the selected tracks
    }
  }
  …
  AliESDv0 *v0=event->GetV0(13);                                    //retrieve the 13th V0
  Int_t ip=v0->GetPositiveIndex(), in=v0->GetNegativeIndex();   //together with its
  AliESDtrack *ptrack=event->GetTrack(ip);                          //positive daughter track
  AliESDtrack *ntrack=event->GetTrack(in);                          //and negative daughter track
  …
```

ESD Example: PID

```
AliESD *event=…;                                            //The reconstructed events are
TTree *esdTree = …;                                         //stored in TTrees  (and so can be “chained”)
Int_t i=0;
while (esdTree->GetEvent(i++) {                             //loop over the reconstructed events
   …                                                        //event selection…

   Double_t priors[AliPID::kSPECIES]={…}                    //A set of a priori probabilities
   AliPID::SetPriors(priors);

   Int_t ntracks=event->GetNumberOfTracks();
   for (i=0; i<ntracks; i++) {                              //loop over ESD tracks (or kinks, V0s …)
       AliESDtrack *track=event->GetTrack(i);

       ULong _t status=AliESDtrack::kTPCpid | AliESDtrack::kTOFpid;
       if ((track->GetStatus()&status) != status)  continue;    //select tracks with the proper status
       if ( … )  continue;                                  //some other selection (quality) criteria


       Double_t probDensity[AliPID::kSPECIES];  track->GetESDpid(probDensity);
       AliPID pid(probDensity);

       Double_t   pp=pid.GetProbablity(AliPID::kProton);    // probability to be a proton
       Double_t   pk=pid.GetProbability(AliPID::kKaon);     // probability to be a kaon
       …
       if (pp > 1./AliPID::kSPECIES) { /* this is a proton */}
   }
}
```

Visualization

- Usage
  - alieve
  - .x alieve_init.C
  - Use then the macros in the EVE folder in TBrowser



```
I-TGeoManager::SetVisLevel: Automatic visible depth disabled
--- Drawing      262 nodes with 3 visible levels
(class Reve::TrackList*)0xad409450
--- Drawing      262 nodes with 3 visible levels
--- Drawing      262 nodes with 3 visible levels
```

Exercises

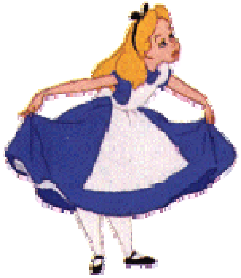
- Event merging

Event mixing – test500

- Generate & reconstruct underlying events (./backgr)
  - Simulation (full chain up to Digits)
    - AliSimulation sim;
    - sim.Run(2);
  - Reconstruction
    - AliReconstruction rec;
    - rec.Run();
- Generate, merge & reconstruct signal events (./signal)
  - Simulation (with event merging)
    - AliSimulation sim;
    - sim.MergeWith(“../backr/galice.root”,3);
    - sim.Run(6);
  - Reconstruction
    - AliReconstruction rec;
    - rec.Run();

Event mixing – test500

```
void test(const char * sdir ="signal",
          const char * bdir ="backgr") {

 TStopwatch timer;
 timer.Start();
 TString name;

 // Signal file, tree, and branch
 name = sdir;
 name += "/AliESDs.root";
 TFile * fSig = TFile::Open(name.Data());
 TTree * tSig = (TTree*)fSig->Get("esdTree");
 TBranch * bSig = tSig->GetBranch("ESD");

 AliESD * esdSig = 0; // The signal ESD object is put here
 bSig->SetAddress(&esdSig);

 // Run loader (signal events)
 name = sdir;
 name += "/galice.root";
 AliRunLoader* rlSig = AliRunLoader::Open(name.Data());

 // Run loader (underlying events)
 name = bdir;
 name += "/galice.root";
 AliRunLoader* rlUnd = AliRunLoader::Open(name.Data(),"Underlying");

 // gAlice
 rlSig->LoadgAlice();
 rlUnd->LoadgAlice();
 gAlice = rlSig->GetAliRun();

 // Now load kinematics and event header
 rlSig->LoadKinematics();
 rlSig->LoadHeader();
 rlUnd->LoadKinematics();
 rlUnd->LoadHeader();

 // Loop on events: check that MC and data contain the same number of events
 Long64_t nevSig = rlSig->GetNumberOfEvents();
 Long64_t nevUnd = rlUnd->GetNumberOfEvents();
 Long64_t nSigPerUnd = nevSig/nevUnd;

 cout << nevSig << " signal events" << endl;
 cout << nevUnd << " underlying events" << endl;
 cout << nSigPerUnd << " signal events per one underlying" << endl;

 for (Int_t iev=0; iev<nevSig; iev++) {
  cout << "Signal event " << iev << endl;
  Int_t ievUnd = iev/nSigPerUnd;
  cout << "Underlying event " << ievUnd << endl;

  // Get signal ESD
  bSig->GetEntry(iev);
  // Get underlying kinematics
  rlUnd->GetEvent(ievUnd);

  // Particle stack
  AliStack * stackSig = rlSig->Stack();
  Int_t nPartSig = stackSig->GetNtrack();
  AliStack * stackUnd = rlUnd->Stack();
  Int_t nPartUnd = stackUnd->GetNtrack();

  Int_t nrec = esdSig->GetNumberOfTracks();
  cout << nrec << " reconstructed tracks" << endl;
  for(Int_t irec=0; irec<nrec; irec++) {
   AliESDtrack * track = esdSig->GetTrack(irec);
   UInt_t label = TMath::Abs(track->GetLabel());
   if (label>=10000000) {
        // Underlying event. 10000000 is the
        // value of fkMASKSTEP in AliRunDigitizer

        label %=10000000;
        if (label>=nPartUnd) continue;
        TParticle * part = stackUnd->Particle(label);

   }
   else {
        cout << " Track " << label << " from the signal event" << endl;
        if (label>=nPartSig) continue;
        TParticle * part = stackSig->Particle(label);
        if(part) part->Print();
   }

  }

 }
 fSig->Close();

 timer.Stop();
 timer.Print();
}
```

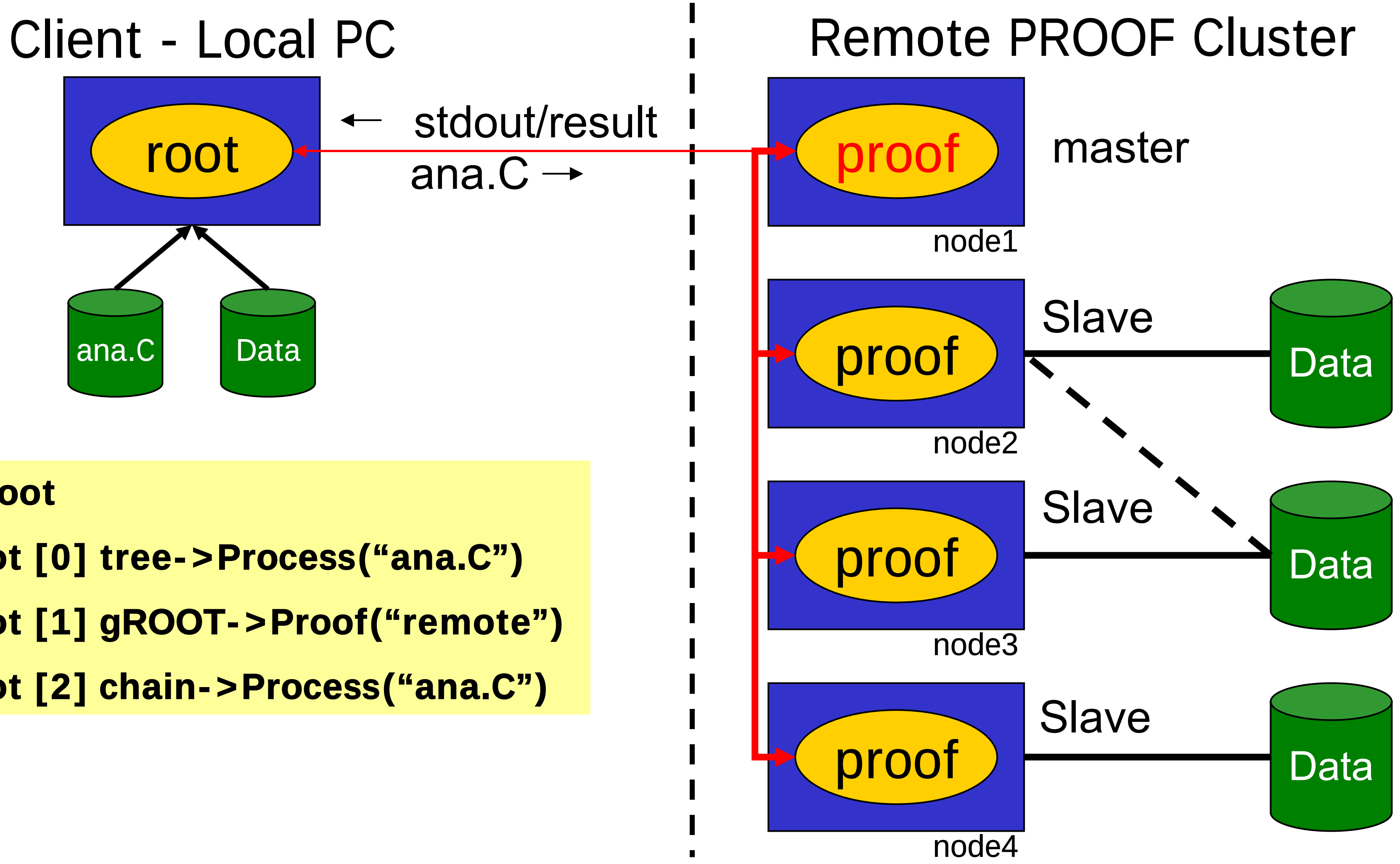


Part II: PROOF

PROOF

- Parallel ROOT Facility
- Interactive parallel analysis on a local cluster
- PROOF itself is not related to Grid
  - Can be used in the Grid
  - Can access Grid files
- The usage of PROOF is transparent
  - The same code can be run locally and in a PROOF system (certain rules have to be followed)
- PROOF is part of ROOT

PROOF Schema



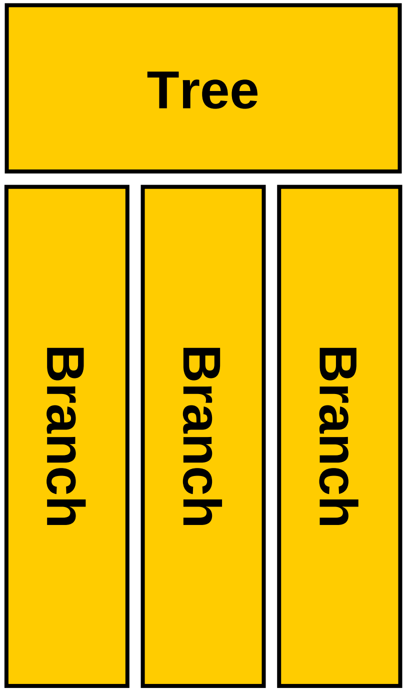
```
$ root
root [0] tree->Process(“ana.C”)
root [1] gROOT->Proof(“remote”)
root [2] chain->Process(“ana.C”)
```

Terminology

- Client
  - Your machine running a ROOT session that is connected to a PROOF master
- Master
  - PROOF machine coordinating work between Slaves
- Slave
  - PROOF machine that processes data
- Query
  - A job submitted from the client to the PROOF system.
    A query consists of a selector and a chain
- Selector
  - A class containing the analysis code (more details later)
- Chain
  - A list of files (trees) to process (more details later)

TTree

- A tree is a container for data storage with disk “overspill”
  - It consists of several *branches*
- These can be in one or several files
- Branches are stored contiguously (split mode)
- When reading a tree, certain branches can be switched off → speed up of analysis when not all data is needed


Tree

Branch

Branch

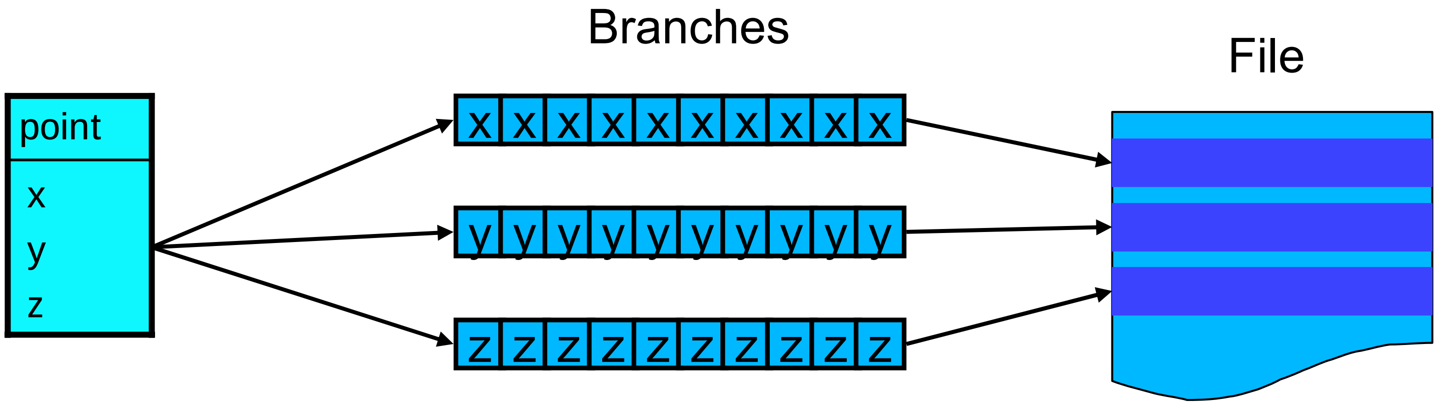
Branch

TTree

```
#include "TTree.h"
#include "TFile.h"
#include "TRandom.h"

class point {
public:
  void Set() {x=gRandom->Rndm();y=gRandom->Rndm();z=gRandom->Rndm();}
private:
  Float_t x, y, z;
  ClassDef(point, 1)
};

Int_t t() {
  point *pp = new point();
  TTree *tree = new TTree("Test","Test Tree",99);
  TFile *file = new TFile("test.root","recreate");
  tree->Branch("point",&pp);
  for(Int_t i=0; i<100; ++i) {
    pp->Set();
    tree->Fill();}
  tree->Write();
  file->Close();
  //
  file=new TFile("test.root","read");
  tree->Print();
  //
  return 0;
}
```

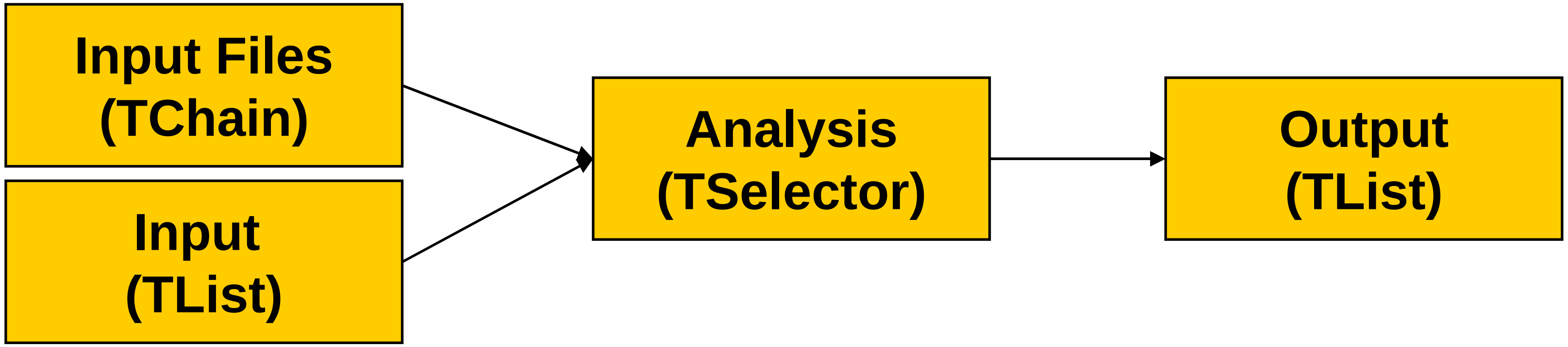
TTree (2)

```
******************************************************************************
*Tree    :Test      : Test Tree                                              *
*Entries :      100 : Total =            4090 bytes  File  Size =          0 *
*        :          : Tree compression factor =   1.00                       *
******************************************************************************
*Branch  :point                                                              *
*Entries :      100 : BranchElement (see below)                              *
*............................................................................*
*Br    0 :x         :                                                        *
*Entries :      100 : Total  Size=       1006 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*............................................................................*
*Br    1 :y         :                                                        *
*Entries :      100 : Total  Size=       1006 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*............................................................................*
*Br    2 :z         :                                                        *
*Entries :      100 : Total  Size=       1006 bytes  One basket in memory    *
*Baskets :        0 : Basket Size=      32000 bytes  Compression=   1.00     *
*............................................................................*
```



How to use PROOF

- Files to be analyzed are put into a chain (→ TChain)
- Analysis written as a selector (→ TSelector, AliSelector, AliSelectorRL)
- Input/Output is sent using dedicated lists
- If additional libraries are needed, these have to be distributed as a “package”


Input Files (TChain)

Input (TList)

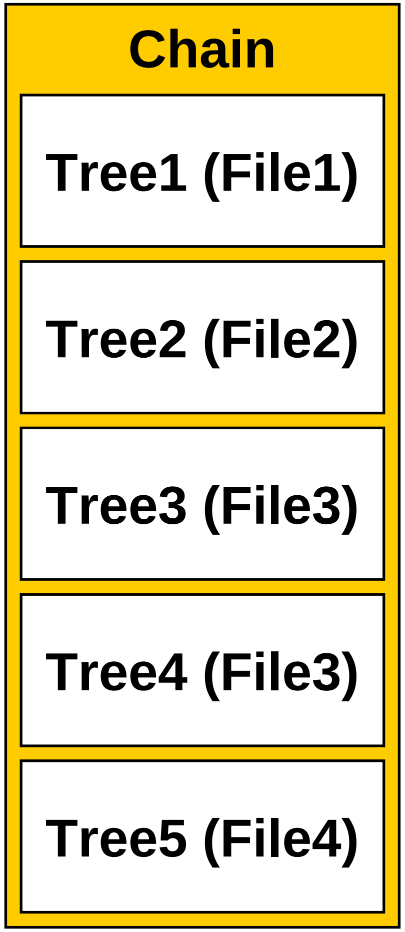
Analysis (TSelector)

Output (TList)

TChain

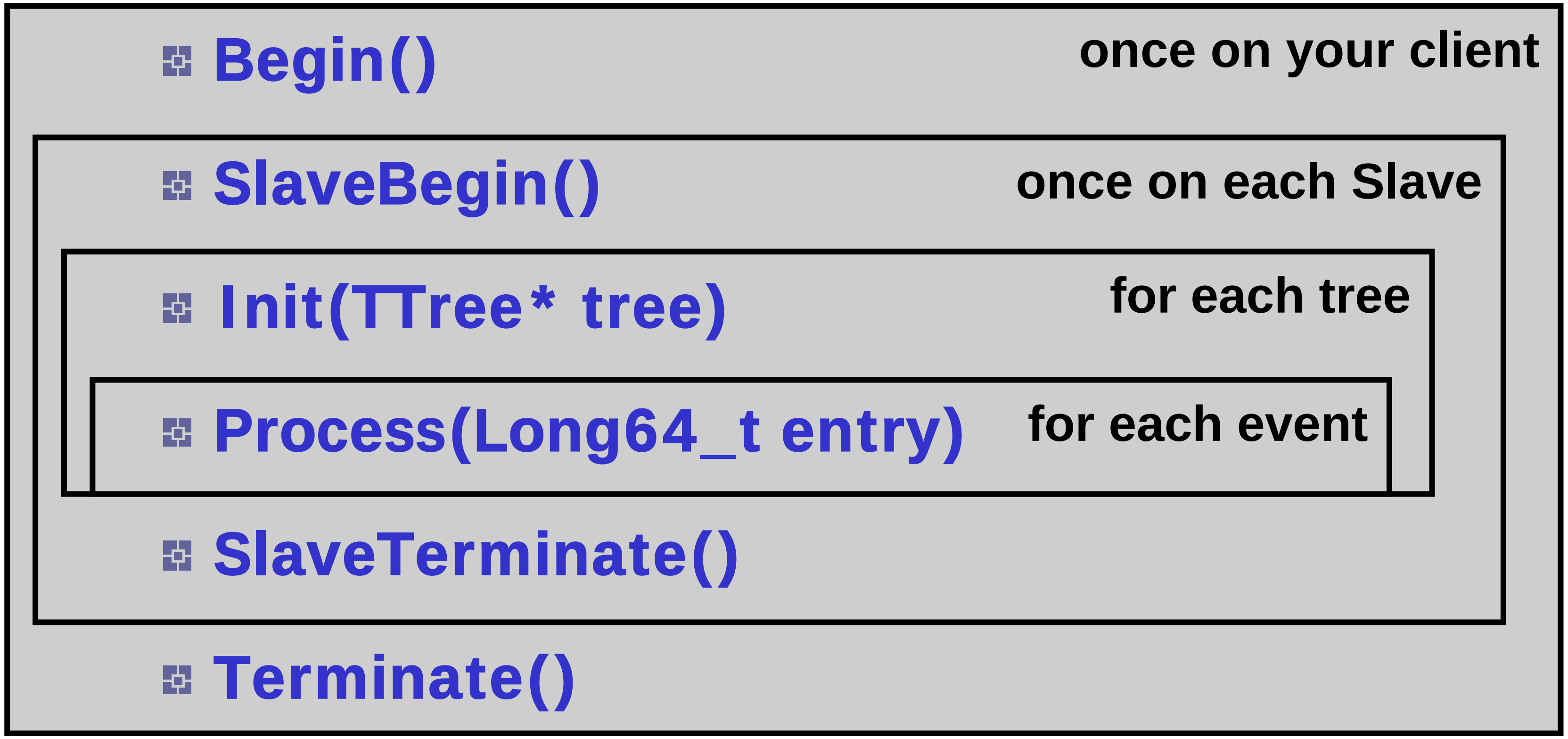
- A chain is a list of trees (in several files)
- Normal TTree functions can be used
  - **Draw(...), Scan(...)**

  → these iterate over all elements of the chain
- Selectors can be used with chains
  - **Process(const char* selectorFileName)**
- After using SetProof() these calls are run in PROOF



TSelector

- Classes derived from TSelector can run locally and in PROOF



- Begin() once on your client
  - SlaveBegin() once on each Slave
    - Init(TTree* tree) for each tree
      - Process(Long64_t entry) for each event
  - SlaveTerminate()
- Terminate()

Input / Output

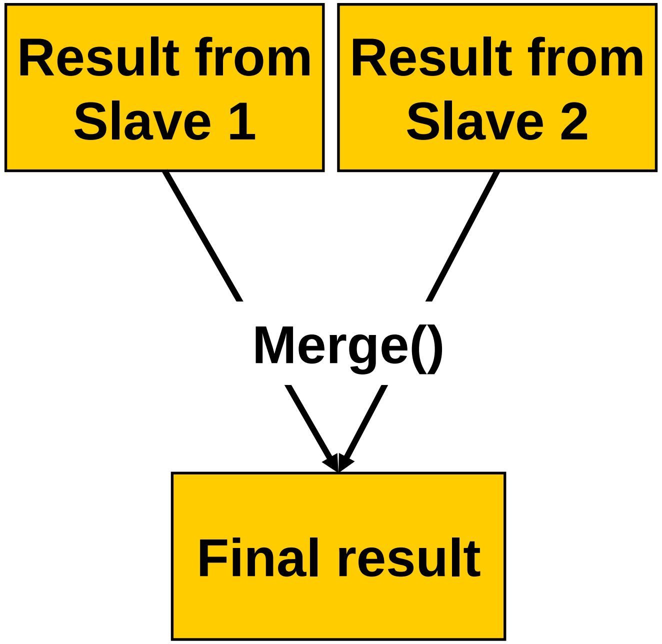
- The TSelector class has two members of type TList:
  - fInput, fOutput
  - These are used to get input data or put output data
- Input list
  - Before running a query the input list is populated **proof->AddInput(myObj)**
  - In the selector (**Begin**, **SlaveBegin**) the object is retrieved: **fInput->FindObject(“myObject”)**

Input / Output (2)

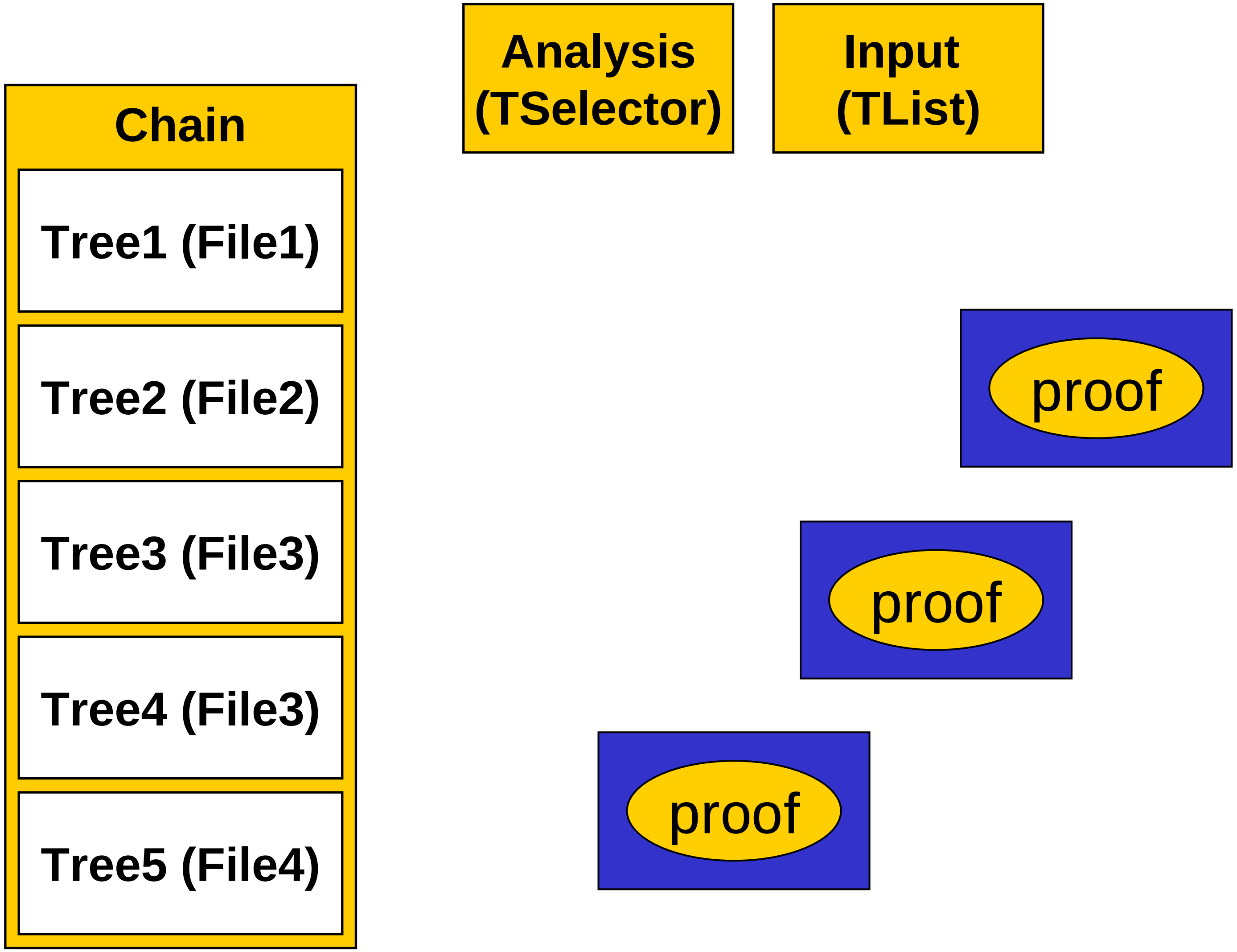
- Output list
  - After processing, the output has to be added to the output list on each Slave (in **SlaveTerminate**) **fOutput->Add(fResult)**
  - PROOF merges the results from each query automatically (see next slide)
  - On your client (in **Terminate**) you retrieve the object and save it, display it, … **fOutput->FindObject("myResult")**

Input / Output (3)

- Merging
  - Objects are identified by name
  - Standard merging implementation for histograms available
  - Other classes need to implement **Merge(TCollection*)**
  - When no merging function is available all the individual objects are returned



Workflow Summary



Chain

Tree1 (File1)

Tree2 (File2)

Tree3 (File3)

Tree4 (File3)

Tree5 (File4)

Analysis (TSelector)

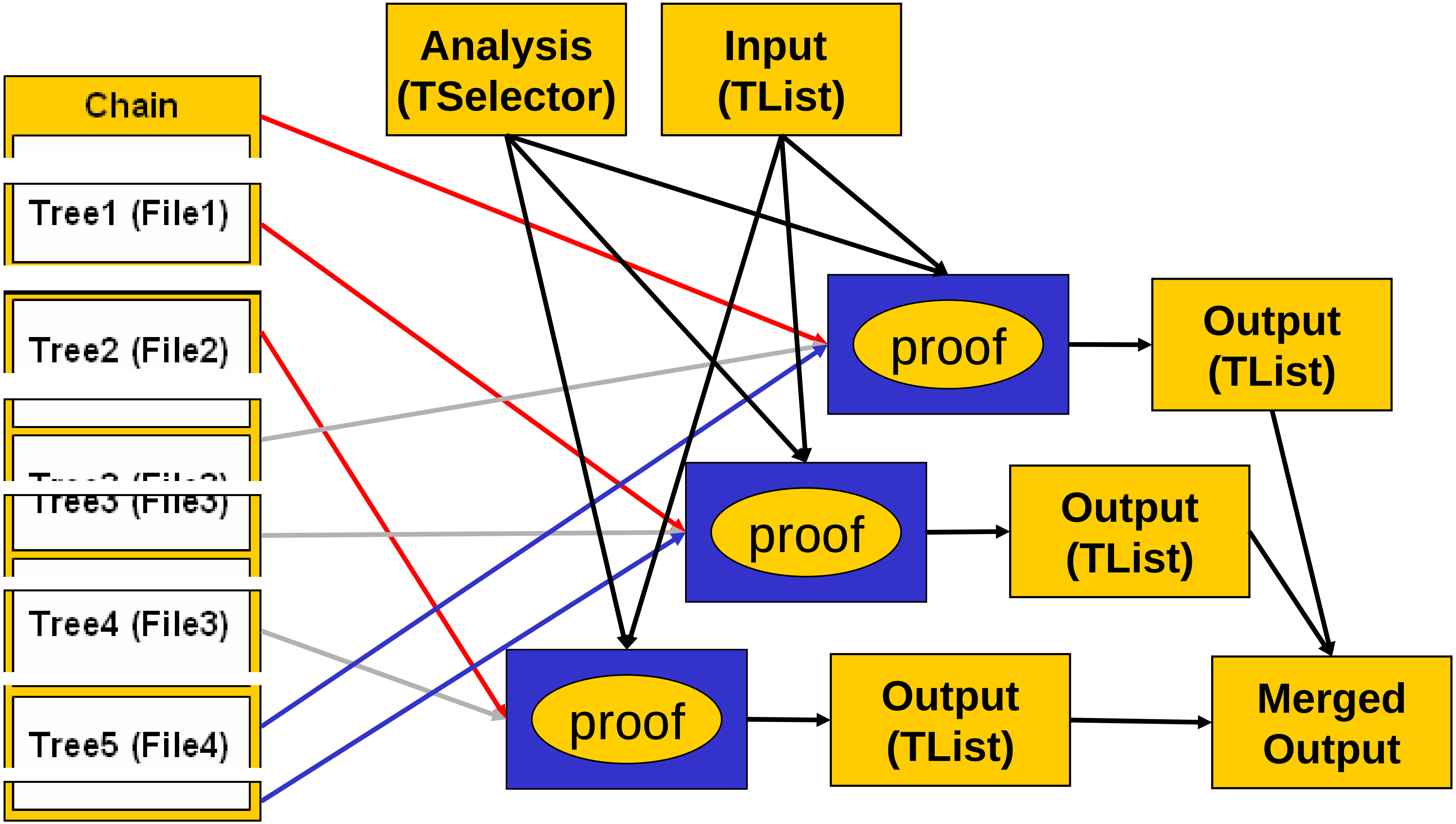
Input (TList)

proof

proof

proof

Workflow Summary



Chain

Tree1 (File1)

Tree2 (File2)

Tree3 (File3)

Tree4 (File3)

Tree5 (File4)

Analysis (TSelector)

Input (TList)

proof

proof

proof

Output (TList)

Output (TList)

Output (TList)

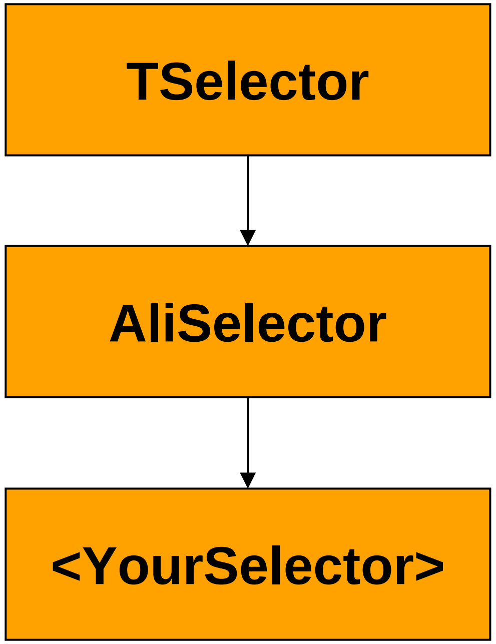
Merged Output

Packages

- PAR files: **P**ROOF **AR**chive. Like Java jar
  - Gzipped tar file
  - PROOF-INF directory
    - BUILD.sh, building the package, executed per Slave
    - SETUP.C, set environment, load libraries, executed per Slave
- API to manage and activate packages
  - **UploadPackage(“package.par”)**
  - **EnablePackage(“package”)**

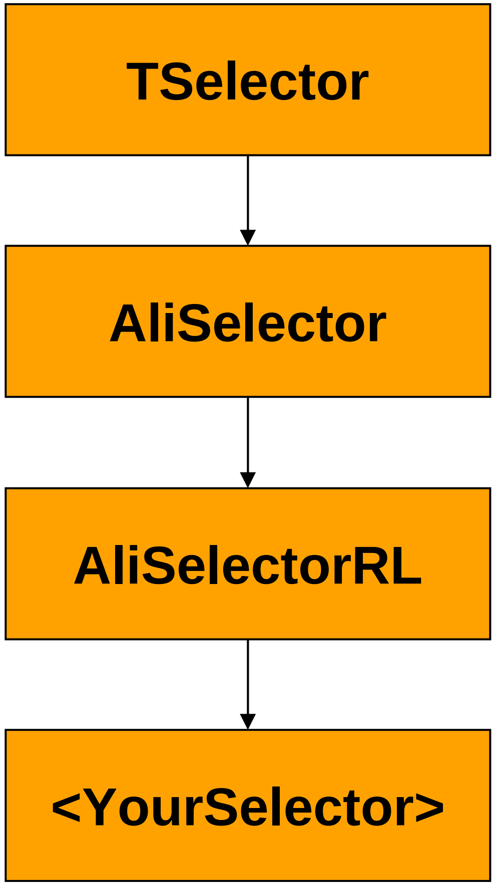
Accessing ESD

- To access AliESDs.root, the ESD.par package has to be uploaded into the PROOF environment
- Selector derives from AliSelector (in STEER)
- Access to data by member: fESD



Accessing the RunLoader

- Access to Kinematics, Clusters, etc. requires access to the RunLoader
- Therefore (nearly) full AliRoot needs to be loaded
- A AliRoot version is already deployed on the CAF test system and can be enabled by a 6 line macro (http://cern.ch/fca/tutorial/proof/ProofEnableAliRoot.C)
- ESD package is not allowed to be loaded
- Selector derives from AliSelectorRL (in STEER)
  - GetStack(), GetRunLoader(), GetHeader()



CERN Analysis Facility

- The **C**ERN **A**nalysis **F**acility (CAF) will run PROOF for ALICE
  - Prompt analysis of pp data
  - Pilot analysis of PbPb data
  - Calibration & Alignment
- Available to the whole collaboration but the number of users will be limited for efficiency reasons
- Design goals
  - 500 CPUs
  - 100 TB of selected data locally available

Evaluation of PROOF

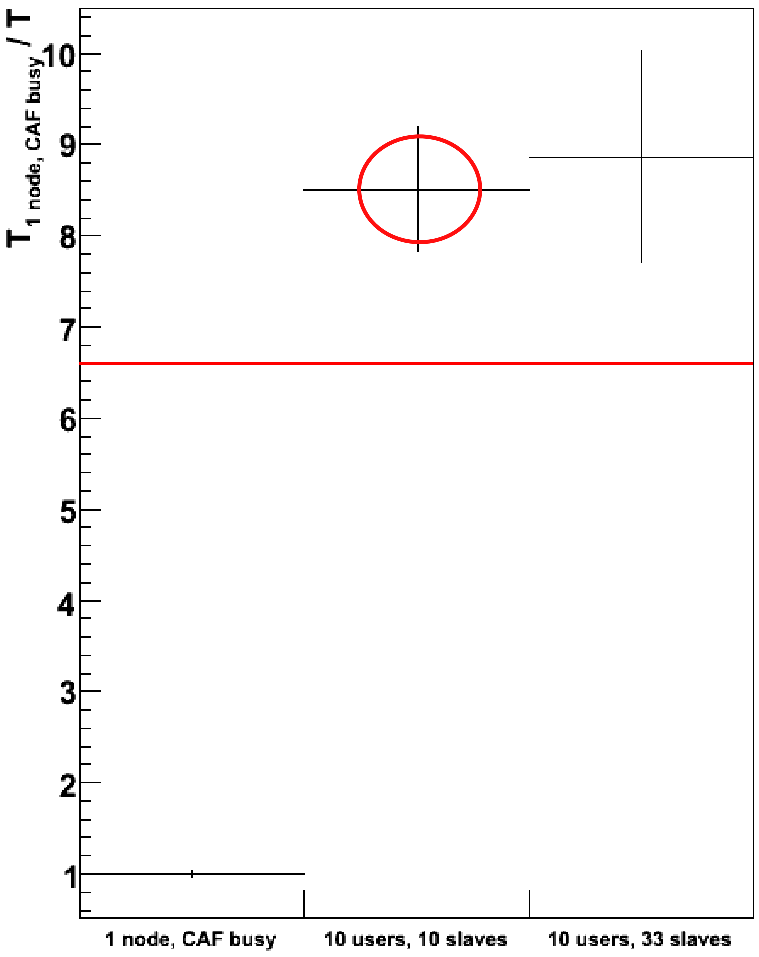
- Test setup since May 2006
  - 40 machines, 2 CPUs each, 200 GB disk
- Tests performed
  - Usability tests
  - Simple speedup plot
  - Evaluation of different query types
  - Evaluation of the system when running a combination of query types
- Goal: Realistic simulation of users using the system

Query Type Cocktail

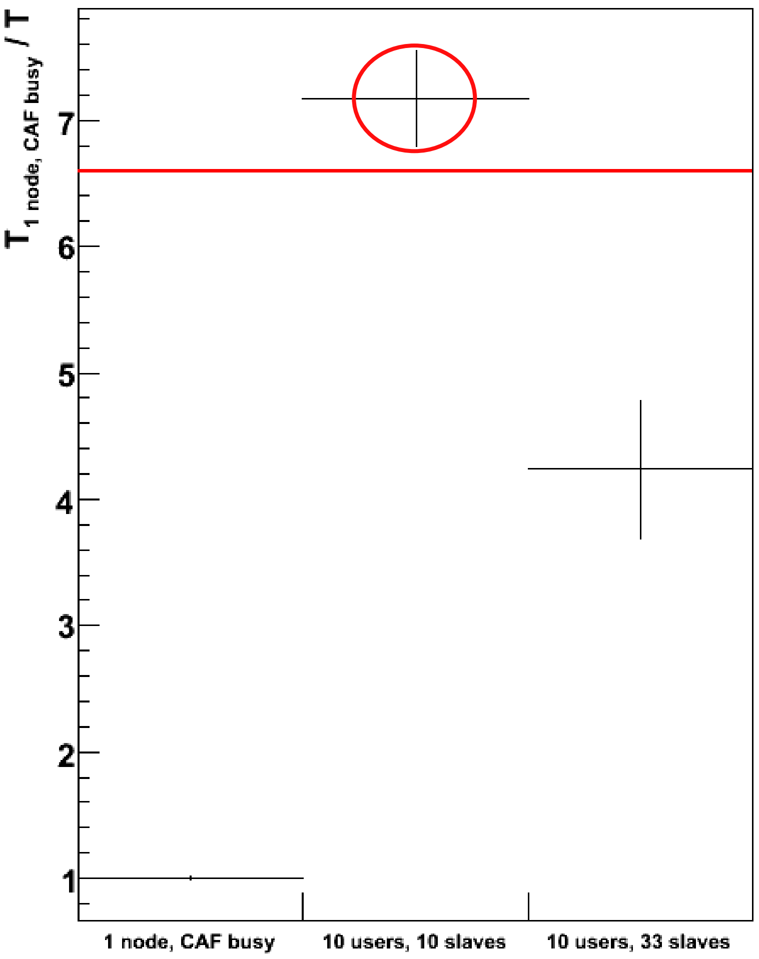
- A realistic stress test consists of different users that submit different types of queries
- 4 different query types
  - 20% very short queries
  - 40% short queries
  - 20% medium queries
  - 20% long queries
- User mix
  - 33 nodes available for the test
  - Maximum average speedup for 10 users = 6.6 (33 nodes = 66 CPUs)

Relative Speedup

Query Short in different environments



Query Medium in different environments



Hands-On

- Getting ready…
- Run selectors that access ESD
  - Locally
  - PROOF
  - Modify it…
- Run selectors that access RunLoader
  - PROOF
  - Modify it…
- Create your own selector

Warm up

- Preconditions
  - Copy files from http://cern.ch/fca/tutorial/proof/tut_proof.tgz to a local directory
- Check AliROOT
  - Start it. Does it show version 5.14/00?
  - $ALICE_ROOT/STEER/AliSelector.h available?
- Add AliROOT include path
  - To ~/.rootrc add
    ACLiC.IncludePaths: -I$(ALICE_ROOT)/include

Files to be used

- CreateESDChain.C Creates a chain from a list of file names
- ESD100_110_v2.txt List of prod2006_2 PDC06 files distributed on the CAF
- ESD.par Par archive for PDC06 data
- ProofEnableAliRoot.C Enables an installed AliROOT on the CAF cluster
- AliMultiplicityESDSelector.{cxx,h} Selector that creates a uncorrected multiplicity histogram from the ESD
- AliMultiplicityMCSelector.{cxx,h} Selector that creates a multiplicity histogram from the MC
- AliEmptySelector.{cxx,h} Empty selector that can be used as a skeleton for your own analysis

Run selector locally

- Start AliRoot
- or… root and

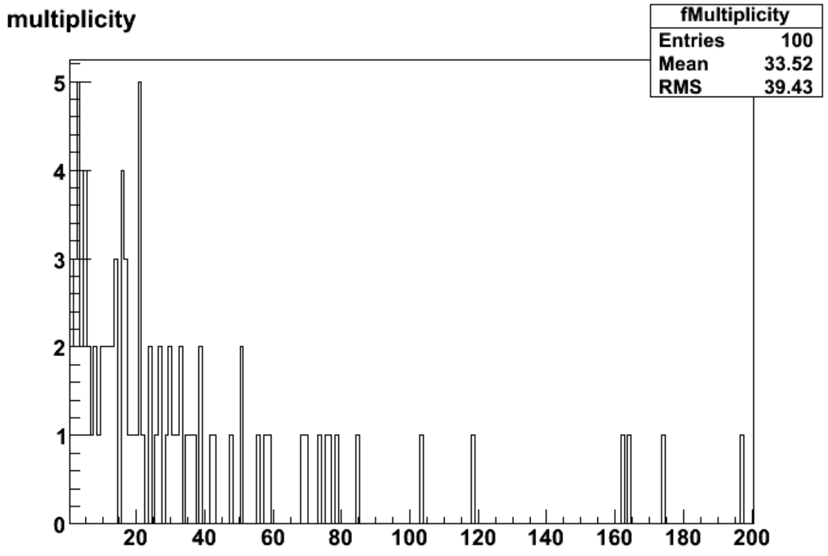
```
root [0] gSystem->Load("libGeom")          // pointer to geometry
root [1] gSystem->Load("libEG")            // pointer to PDG database
root [2] gSystem->Load("libESD")           // ESD library
```

- Create a chain

```
chain = new TChain("esdTree")
chain->Add("root://lxb6046.cern.ch//pool/proofpool/pdc06/100/002/root_archive.zip#AliESDs.root")
chain->GetEntries()                        // Should return 100
```

- Execute a selector locally

```
chain->Process("AliMultiplicityESDSelector.cxx+")
```



Run selector with PROOF

- Start ROOT or AliRoot and create chain as before
- Connect to PROOF server

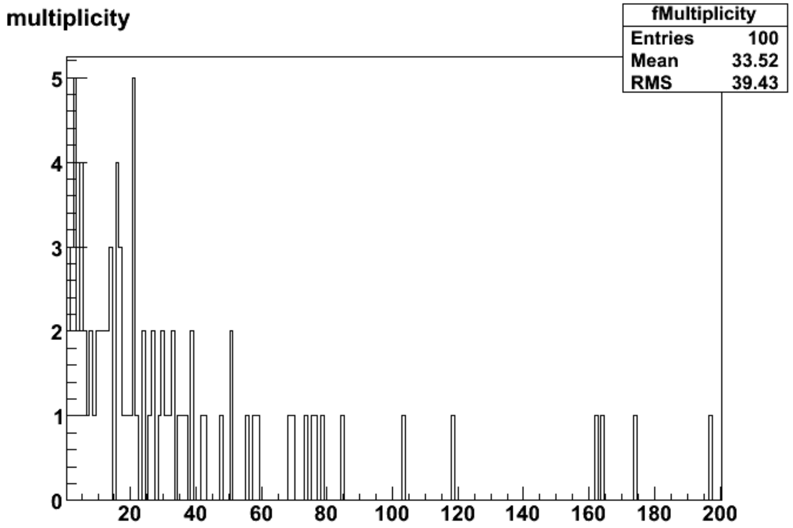
```
proof = TProof::Open(“<username>@lxb6046”)
```

- Upload the ESD package

```
proof->UploadPackage(“ESD.par”)
proof->EnablePackage(“ESD”)
```

- Execute with PROOF

```
chain->SetProof()
chain->Process(“Ali
      Multiplicity
      ESDSelector.cxx+”)
```



Run with a long chain

- Create longer chain

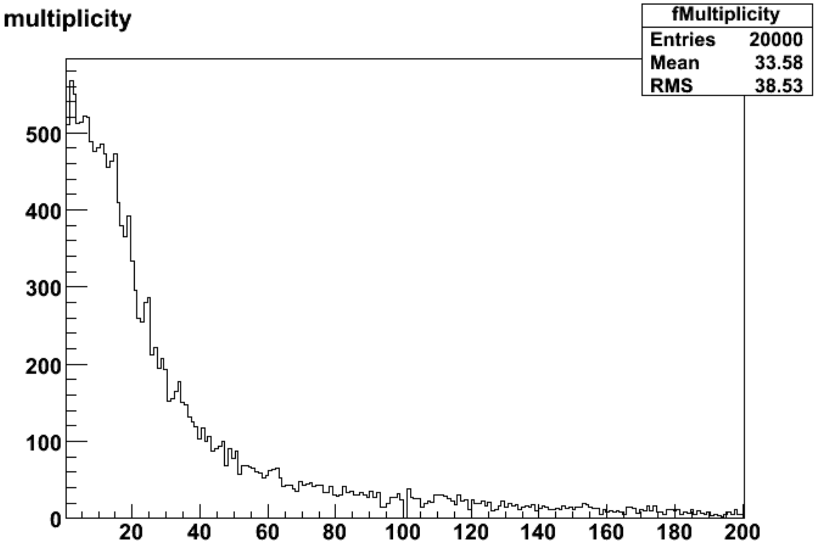
```
chain = new TChain("esdTree")
int count=0; TString file; ifstream in;
in.open("ESD100_110_v2.txt")
while((++count<201) && (in>>file)) chain->Add(file.Data());
```

  - Alternative

```
.L CreateESDChain.C
chain = CreateESDChain.C(“ESD100_110_v2.txt”)
```

- Execute the selector with PROOF

```
chain->SetProof()
chain->Process(“AliMultiplicity
    ESDSelector.cxx+”)
```



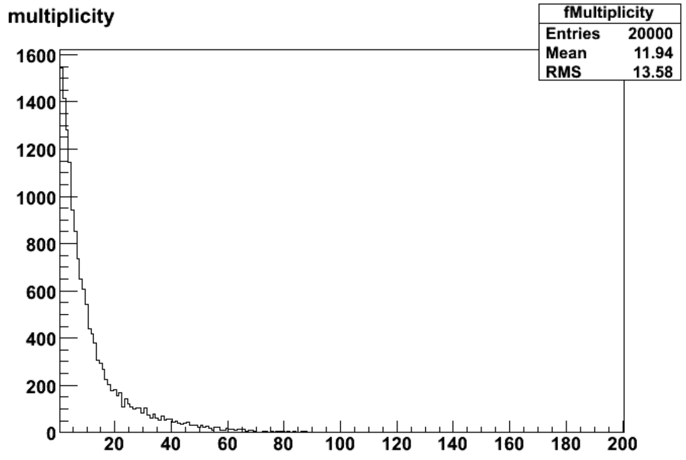
Looking at the Selector

- SlaveBegin
  - Called once per Slave before processing
  - Multiplicity histogram is created
- Process
  - Called once per event
  - Tracks are counted, histogram filled
- SlaveTerminate
  - Called once per Slave after processing
  - Multiplicity histogram is filled into the output list
- Terminate
  - Called once on the client (your laptop/PC)
  - Multiplicity histogram is read from the output list and displayed

Changing the Selector

- Add a |η| < 0.5 cut

```
Int_t nGoodTracks = 0;
for (Int_t i=0; i<fESD->GetNumberOfTracks(); ++i)
{
  AliESDtrack* track = fESD->GetTrack(i);
  Double_t p[3];
  TVector3 vector(p);
  track->GetConstrainedPxPyPz(p);
  Float_t eta = vector.Eta();
  if (TMath::Abs(eta) < 0.5)
    nGoodTracks++;
}
fMultiplicity->Fill(nGoodTracks);
```



Changing the Selector (2)

- Add a second plot: η distribution
  - Header file (.h file)
    - Add new member: TH1F* fEta;   // eta distribution
  - Constructor
    - Initialize member: fEta(0)
  - SlaveBegin
    - Create histogram
      fEta = new TH1F("fEta", "#eta distribution", 20, -2, 2);
  - Process
    - Get η like in previous example
    - Fill histogram: fEta->Fill(eta);

Changing the Selector (3)

- SlaveTerminate
  - Add histogram to the output list: `fOutput->Add(fEta);`
- Terminate
  - Read histogram from the output list

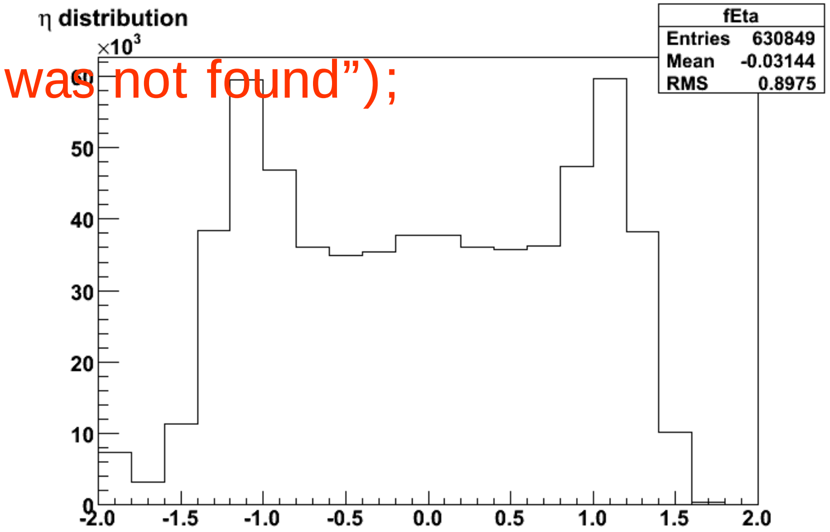
```
fEta = dynamic_cast<TH1F*> (fOutput->FindObject("fEta"));
```

  - Introduce an if statement if the object was retrieved

```
if (!fEta)  {
  AliDebug(AliLog::kError, “fEta was not found”);
  return;
}
```

  - Draw the histogram

```
new TCanvas;
fEta->DrawCopy();
```



Learning about Branches

- The ESD tree consists of several branches
- Switching off not needed branches increases speed of analysis significantly
- Looking at the available branches

```
chain = new TChain(“esdTree”)
chain->Add(“root://lxb6046.cern.ch//pool/proofpool/
    pdc06/100/002/root_archive.zip#AliESDs.root”)
chain->Print()
```

- Disable all branches (in Init)

```
tree->SetBranchStatus("*", 0)
```

- Enable a needed branch (in Init)

```
tree->SetBranchStatus(“fTracks”, 1)
```

- Try this! What is the increase in processing speed?

Running with full AliROOT

- Restart ROOT session
- Connect to PROOF server

```
proof = TProof::Open(“<username>@lxb6046”)
```

- Enable AliROOT

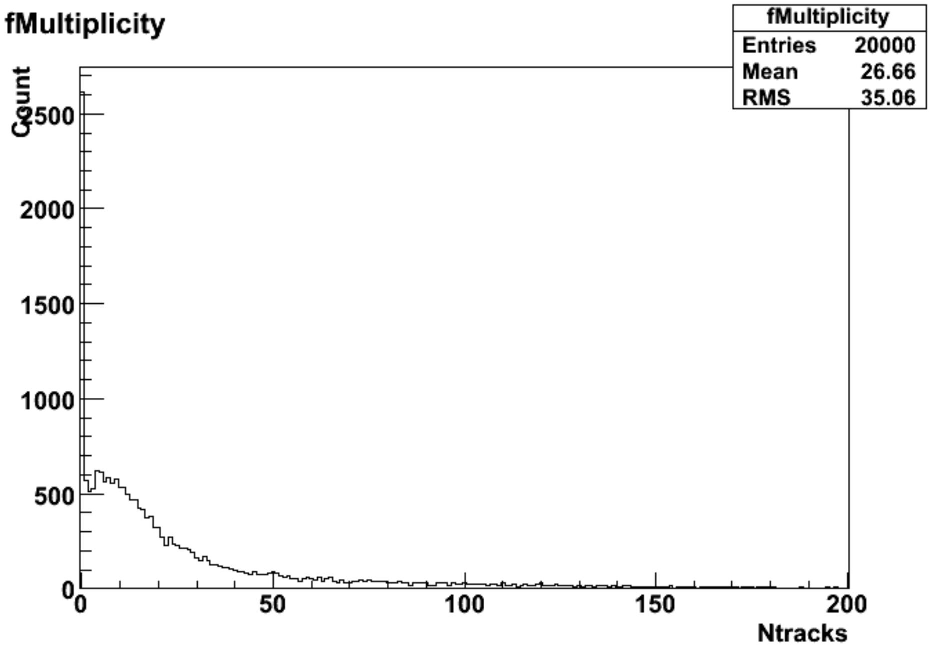
```
.x ProofEnableAliRoot.C
```

- Create Chain

```
.L CreateESDChain.C
chain = CreateESDChain(
   “ESD100_110_v2.txt”)
```

- Execute the selector that accesses MC

```
chain->SetProof()
chain->Process(“AliMultiplicit
   MCSelector.cxx+”)
```



Accessing the event header

- AliSelectorRL::GetHeader returns the header
- Retrieve the header (in Process)

```
AliHeader* header = GetHeader();
if (!header)
{
  AliDebug(AliLog::kError, "Header not available");
  return kFALSE;
}
```

- Retrieve a value from the header

```
printf("This is run %d.\n", header->GetRun());
```

- Run it and look at the log

Your own selector

- Start from AliEmptySelector
- Find a name

  Copy AliEmptySelector.h/.cxx to <yourSelector>.h/.cxx

  - Replace class names, define statement
- Put in your analysis code

Reading log files

- When your selector crashes
  - You cannot access the output via the PROOF progress window
  - Usually you have to restart the ROOT session
- Reading output from last query
  - Open ROOT
  - Get a PROOF manager object
    ```
    mgr = TProof::Mgr(“<username>@lxb6046”)
    ```
  - Get the log files from the last session
    ```
    logs = mgr->GetSessionLogs(0)
    ```
  - Display them
    ```
    logs->Display()
    ```
  - Search for a special word (e.g. segmentation violation)
    ```
    logs->Grep(“segmentation violation”)
    ```
  - Save them to a file
    ```
    logs->Save(“*”, “logs.txt”)
    ```

Some Goodies...

- Resetting environment
  - TProof::Reset("<username>@lxb6046")
- Run with debug
  - Process("<selector>", "debug")
  - Process("<selector>", "moredebug")
- Compile with debug
  - Process("<selector>+g")
- Create a package from AliROOT
  - make ESD.par

Backup

PROOF Handout

- Connect to PROOF server

```
  proof = TProof::Open(“<username>@lxb6046”)
```

- Upload the ESD package

```
  proof->UploadPackage(“ESD.par”)
  proof->EnablePackage(“ESD”)
```

- Enable AliROOT

```
  .x ProofEnableAliRoot.C
```

- Create small chain manually

```
  chain = new TChain(“esdTree”)
  chain->Add(“root://lxb6046//proofpool/pdc06/100/
      002/root_archive.zip#AliESDs.root”)
```

- Create long chain

```
.L CreateESDChain.C
chain = CreateESDChain.C(“ESD100_110_v2.txt”)
```

- Execute a selector with PROOF

```
  chain->SetProof()
  chain->Process(“AliMultiplicityESDSelector.cxx+”)
```

Analysis with Selectors

- Create a selector

```
TFile *fESD = TFile::Open("AliESDs.root");
TTree *tESD = (TTree *)fESD->Get("esdTree");
tESD->MakeSelector();
Info in <TTreePlayer::MakeClass>: Files: esdTree.h and esdTree.C generated from
    TTree: esdTree
```

- Modify the selector accordingly and run it

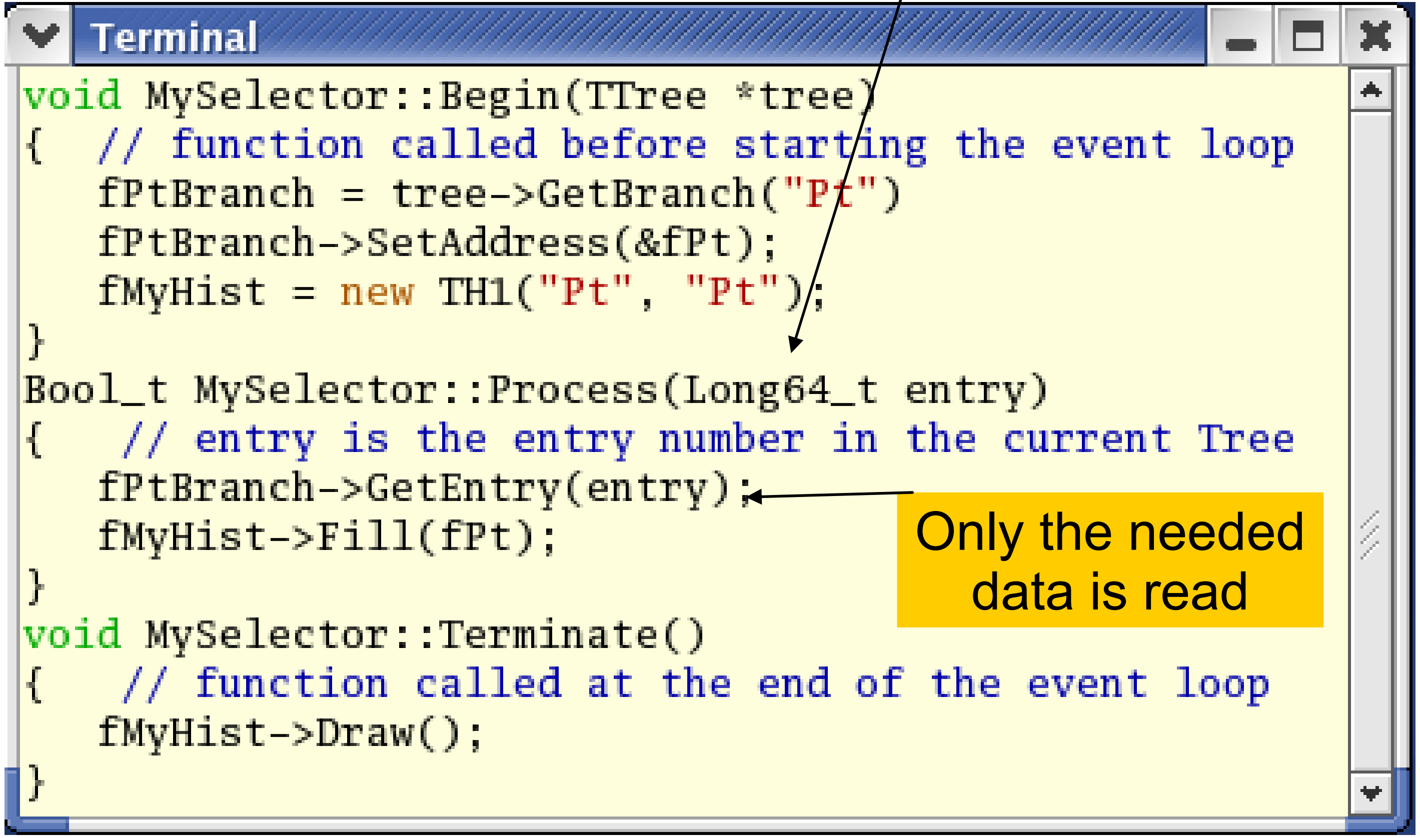
```
.! emacs esdTree.{C,h}&
tESD->Process("esdTree.C");
```

- Use an existing example and modify it accordingly
- For further information on selectors:
  - Check the analysis user guide provided by A.Peters
  - Presentation of M.Biskup at the Root workshop 2005

Basics of Selectors

- Selectors contain skeleton of processing system
  - Preprocessing and initialization
  - Processing each event
  - Post processing and clean-up

Entries are processed in an arbitrary order



```
void MySelector::Begin(TTree *tree)
{   // function called before starting the event loop
    fPtBranch = tree->GetBranch("Pt")
    fPtBranch->SetAddress(&fPt);
    fMyHist = new TH1("Pt", "Pt");
}
Bool_t MySelector::Process(Long64_t entry)
{    // entry is the entry number in the current Tree
    fPtBranch->GetEntry(entry);
    fMyHist->Fill(fPt);
}
void MySelector::Terminate()
{    // function called at the end of the event loop
    fMyHist->Draw();
}
```

Skeleton can be generated from a Tree

Only the needed data is read

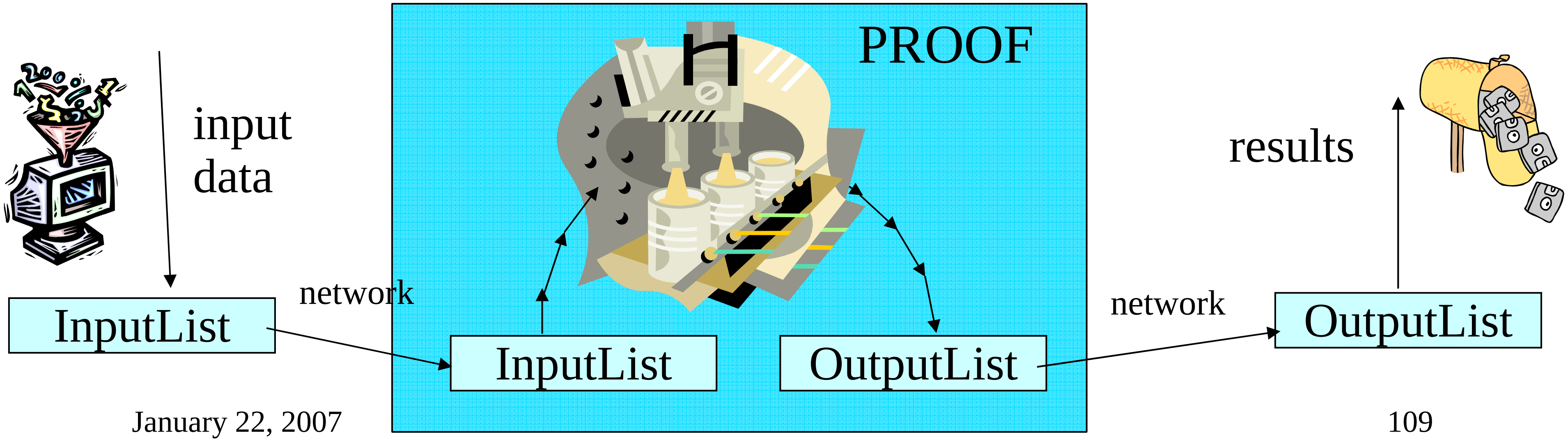
Selectors

- More complicated in a distributed environment
  - Many computers to initialize and clean-up
  - Many trees in a chain
  - Input and output results should be transparently sent over network

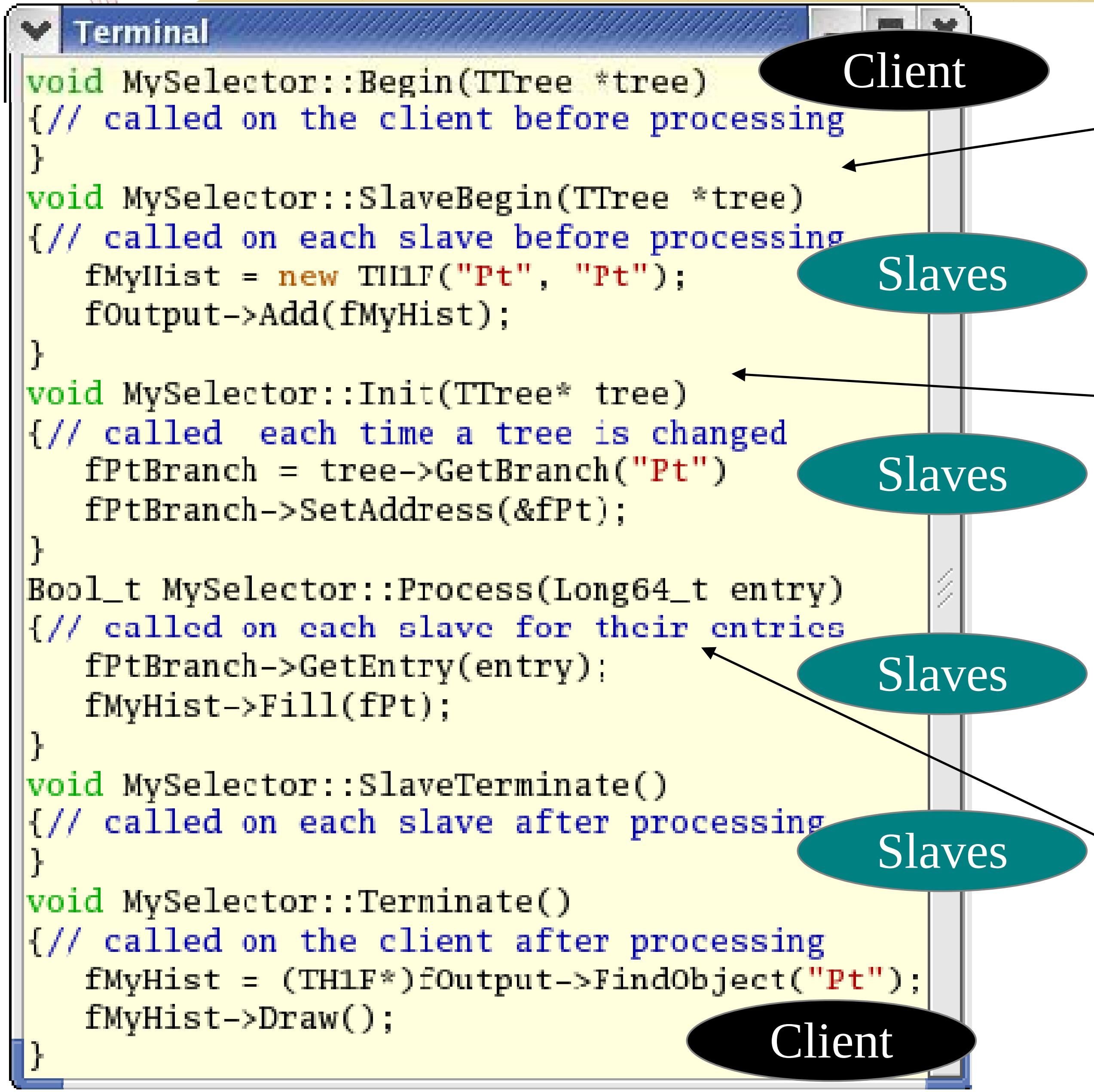
```
TSelector::SlaveBegin();
TSelector::SlaveTerminate();
```

```
TSelector::Init(TTree*)
```

```
TList* fInput, fOutput;
```



PROOF and Selectors



```
void MySelector::Begin(TTree *tree)
{// called on the client before processing
}
void MySelector::SlaveBegin(TTree *tree)
{// called on each slave before processing
   fMyHist = new TH1F("Pt", "Pt");
   fOutput->Add(fMyHist);
}
void MySelector::Init(TTree* tree)
{// called  each time a tree is changed
   fPtBranch = tree->GetBranch("Pt")
   fPtBranch->SetAddress(&fPt);
}
Bool_t MySelector::Process(Long64_t entry)
{// called on each slave for their entries
   fPtBranch->GetEntry(entry);
   fMyHist->Fill(fPt);
}
void MySelector::SlaveTerminate()
{// called on each slave after processing
}
void MySelector::Terminate()
{// called on the client after processing
   fMyHist = (TH1F*)fOutput->FindObject("Pt");
   fMyHist->Draw();
}
```

Initialize each Slave

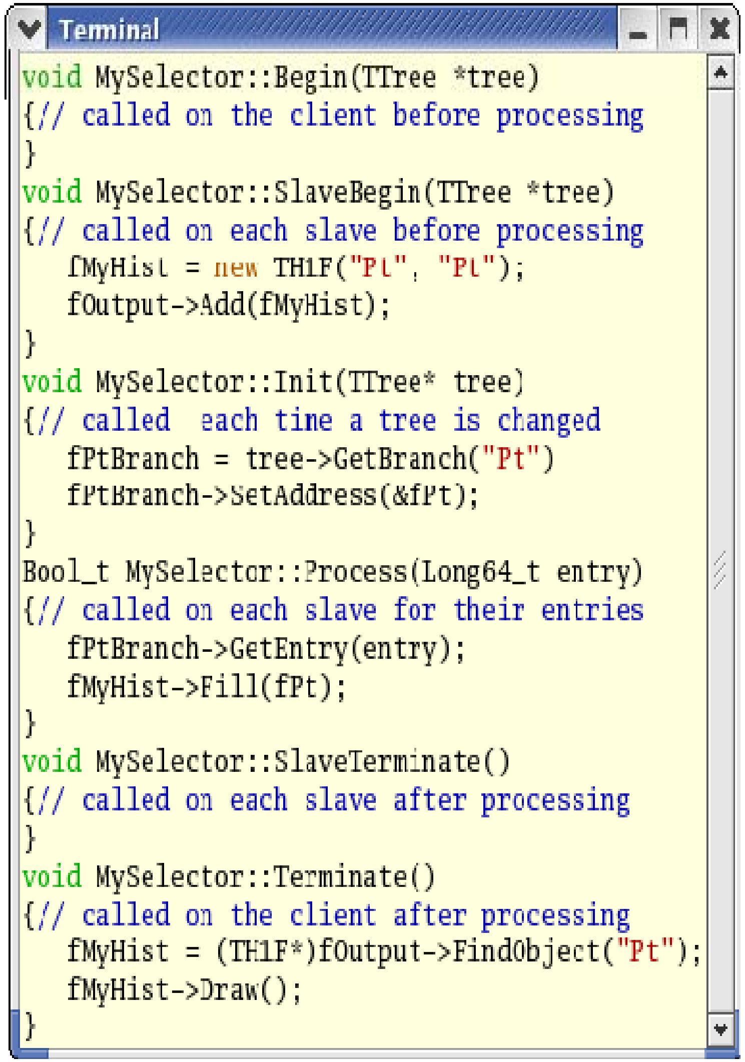
Many Trees are being processed

The same code works also without PROOF (of course!)

No user's control on the order

Selectors - summary

- Skeletons generated from a tree
- Only methods need to be filled
- Simplify program structure
- Can be used for parallel processing as well as for local analysis
- More about the selectors during the PROOF tutorial



```
void MySelector::Begin(TTree *tree)
{// called on the client before processing
}
void MySelector::SlaveBegin(TTree *tree)
{// called on each slave before processing
   fMyHist = new TH1F("Pt", "Pt");
   fOutput->Add(fMyHist);
}
void MySelector::Init(TTree* tree)
{// called  each time a tree is changed
   fPtBranch = tree->GetBranch("Pt")
   fPtBranch->SetAddress(&fPt);
}
Bool_t MySelector::Process(Long64_t entry)
{// called on each slave for their entries
   fPtBranch->GetEntry(entry);
   fMyHist->Fill(fPt);
}
void MySelector::SlaveTerminate()
{// called on each slave after processing
}
void MySelector::Terminate()
{// called on the client after processing
   fMyHist = (TH1F*)fOutput->FindObject("Pt");
   fMyHist->Draw();
}
```

PROOF Installation

- Install ROOT with PROOF enabled (default)
  - More information: http://root.cern.ch
- Configuration (see next slides)
  - xrootd config file: xrd.cf
  - PROOF config file: proof.conf
- Start xrootd service
  - Requires unprivileged user account

xrd.cf

```
## Load the XrdProofd protocol:
xrd.protocol xproofd:1093 /opt/root/lib/libXrdProofd.so
## Set ROOTSYS
xpd.rootsys /opt/root
## Working directory for sessions
xpd.workdir /pool/proofbox
```

xrd.cf (2)

```
## xpd.resource static [<cfg_file>] [ucfg:<user_cfg_opt>]  [wmx:<max_workers>] [selopt:<selection_mode>]
xpd.resource static /etc/proof/proof.conf wmx:-1 selopt:roundrobin
## Server role (master, worker) [default: any]
xpd.role worker if lxb*.cern.ch
xpd.role master if lxb6046.cern.ch
## Master(s) allowed to connect. By default all connections are allowed.
xpd.allow lxb6046.cern.ch
```

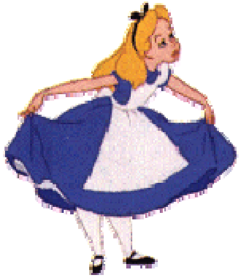
proof.conf

```
## machine running the master
master lxb6046.cern.ch
## machine(s) running Workers, dual CPU machines have to be listed twice
worker lxb6047.cern.ch
worker lxb6047.cern.ch
worker lxb6048.cern.ch
worker lxb6048.cern.ch
…
```

Starting xrootd Service

```
xrootd -b -l xrootd.log -R proofaccount -c xrd.cf -d
```

- Options:
  - -b : background (skip for debugging)
  - -l : log file
  - -R <useraccount> : user account that runs xrootd service
  - -c <configfile> : configuration file
  - -d : debug flag
- Do not forget full paths to the files



Part III: AliEn

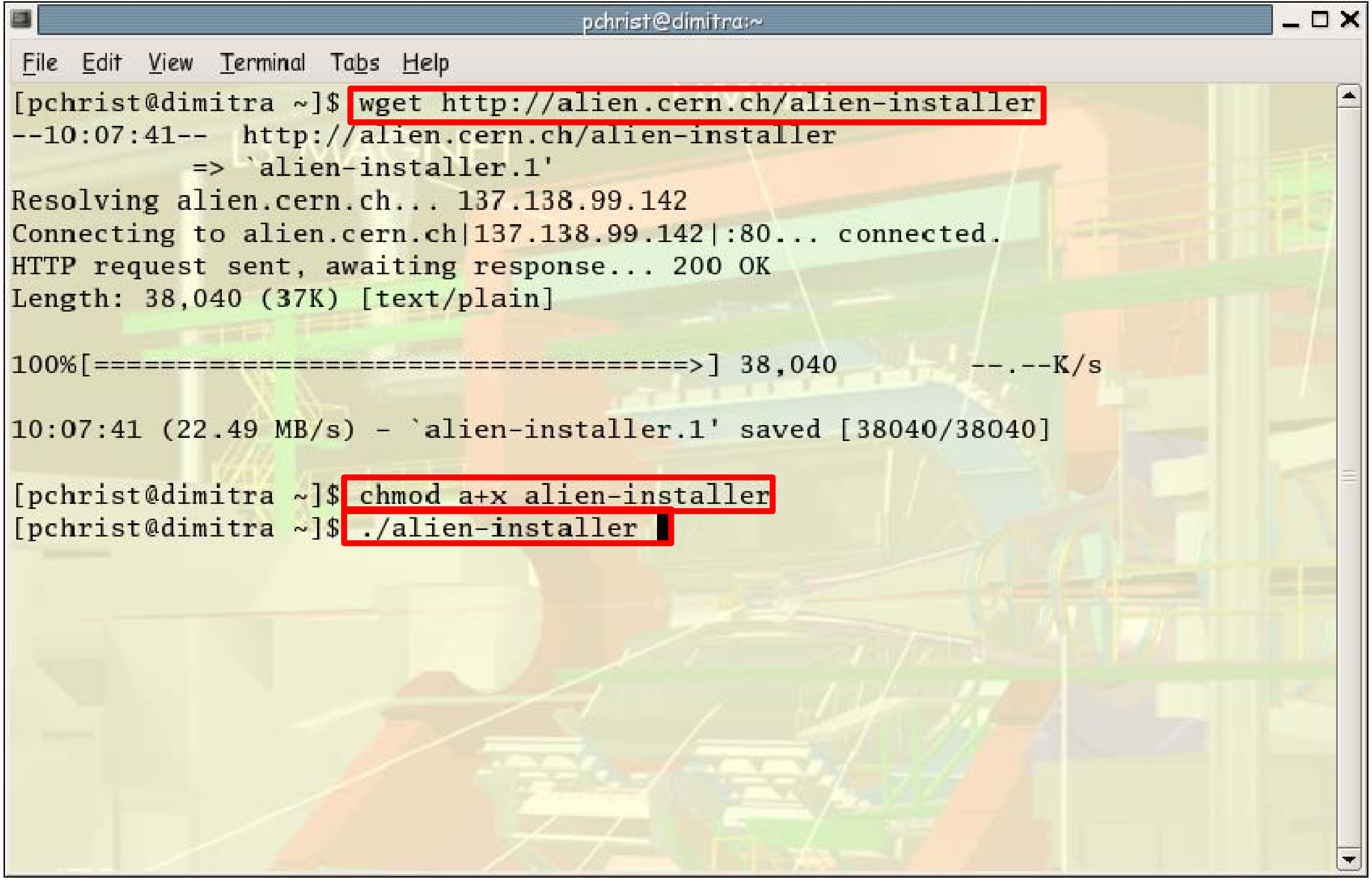
Outline I

- Installation of the AliEn software.
- Authentication – Possible problems.
- General description of the shell:
  - Basic commands.
  - Basic functionalities.
- Working with the file catalogue:
  - Copying files from/to the catalogue.
  - File catalogue structure.
  - Querying the file catalogue.
- ROOT API
- News on production.

Outline II

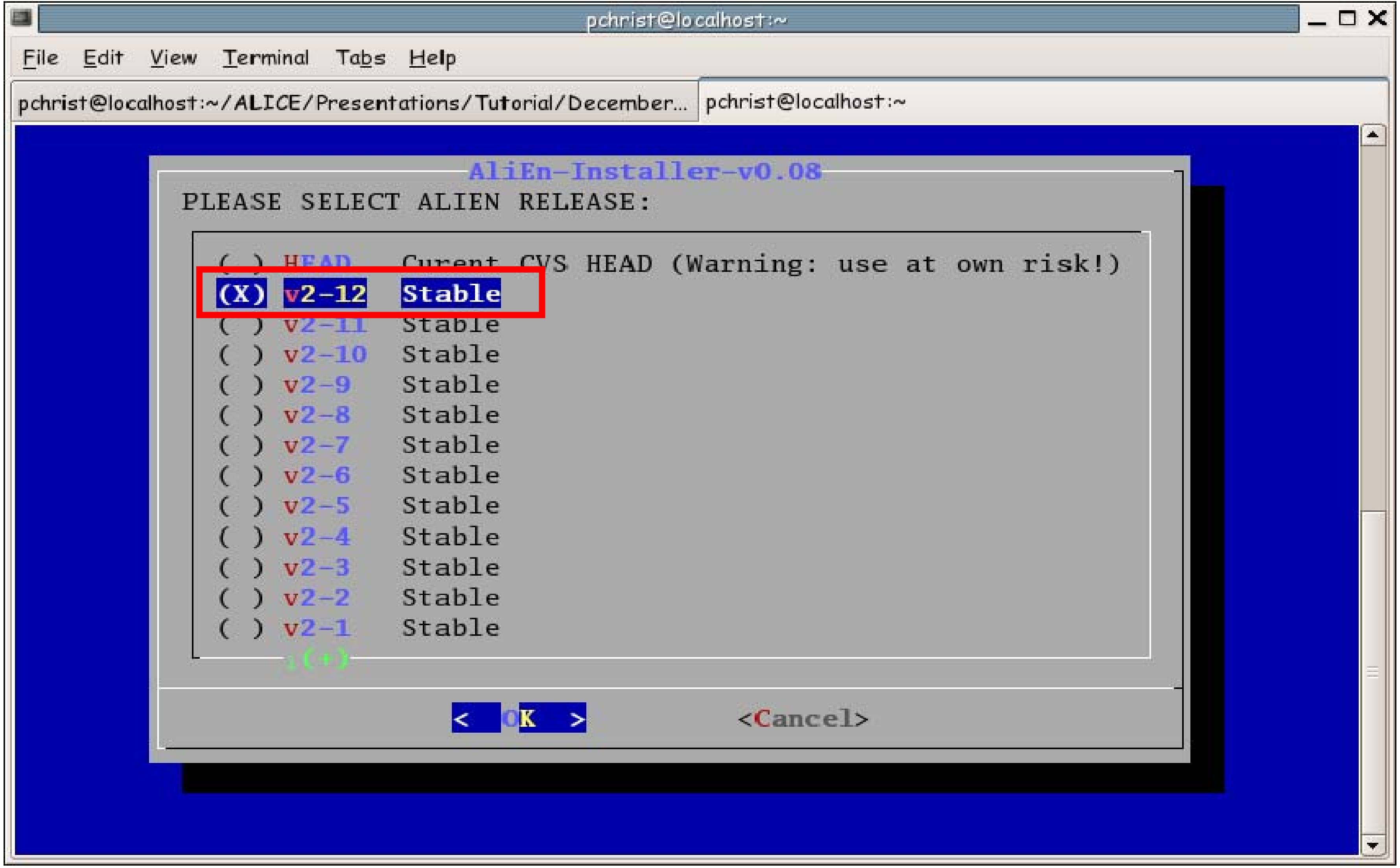
- Flow of the overall analysis procedure.
- New analysis framework
- Local analysis:
  - Creation of tag files.
  - Local analysis using the Event Tag System.
- Interactive analysis with AliEn stored files.
- Batch analysis:
  - Flow of the procedure.
  - Description of the files needed.
  - Description of the jdl fields.
  - Practical examples.

Installation – Getting the installer



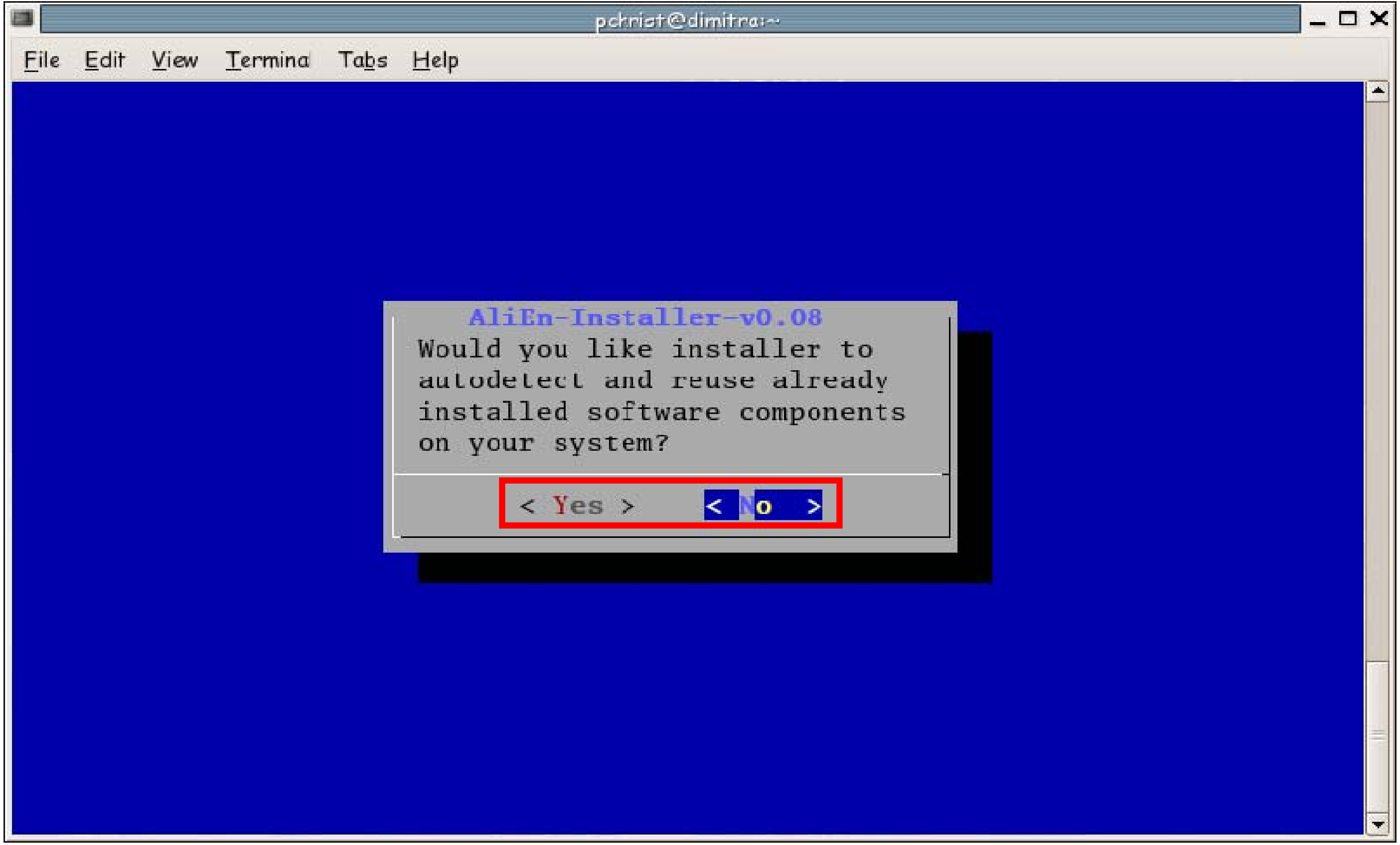
```
[pchrist@dimitra ~]$ wget http://alien.cern.ch/alien-installer
--10:07:41--  http://alien.cern.ch/alien-installer
           => `alien-installer.1'
Resolving alien.cern.ch... 137.138.99.142
Connecting to alien.cern.ch|137.138.99.142|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 38,040 (37K) [text/plain]

100%[====================================>] 38,040        --.--K/s

10:07:41 (22.49 MB/s) - `alien-installer.1' saved [38040/38040]

[pchrist@dimitra ~]$ chmod a+x alien-installer
[pchrist@dimitra ~]$ ./alien-installer
```

Installation – Selecting the version



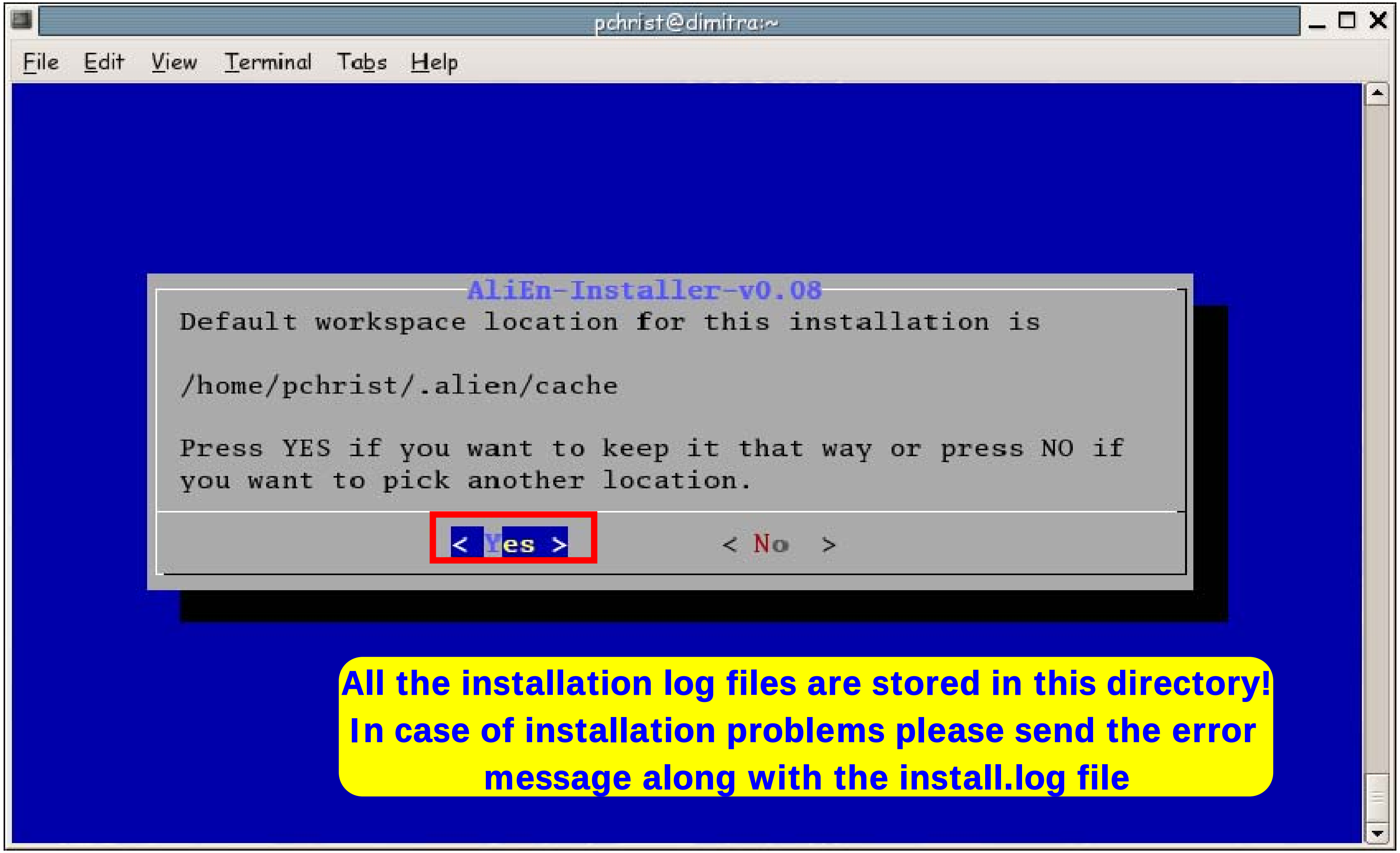
```
AliEn-Installer-v0.08
PLEASE SELECT ALIEN RELEASE:

 ( ) HEAD    Curent CVS HEAD (Warning: use at own risk!)
 (X) v2-12   Stable
 ( ) v2-11   Stable
 ( ) v2-10   Stable
 ( ) v2-9    Stable
 ( ) v2-8    Stable
 ( ) v2-7    Stable
 ( ) v2-6    Stable
 ( ) v2-5    Stable
 ( ) v2-4    Stable
 ( ) v2-3    Stable
 ( ) v2-2    Stable
 ( ) v2-1    Stable
       ↓(+)

          <  OK  >           <Cancel>
```

Installation – Reuse of software



```
AliEn-Installer-v0.08
Would you like installer to
autodetect and reuse already
installed software components
on your system?

< Yes >        < No >
```

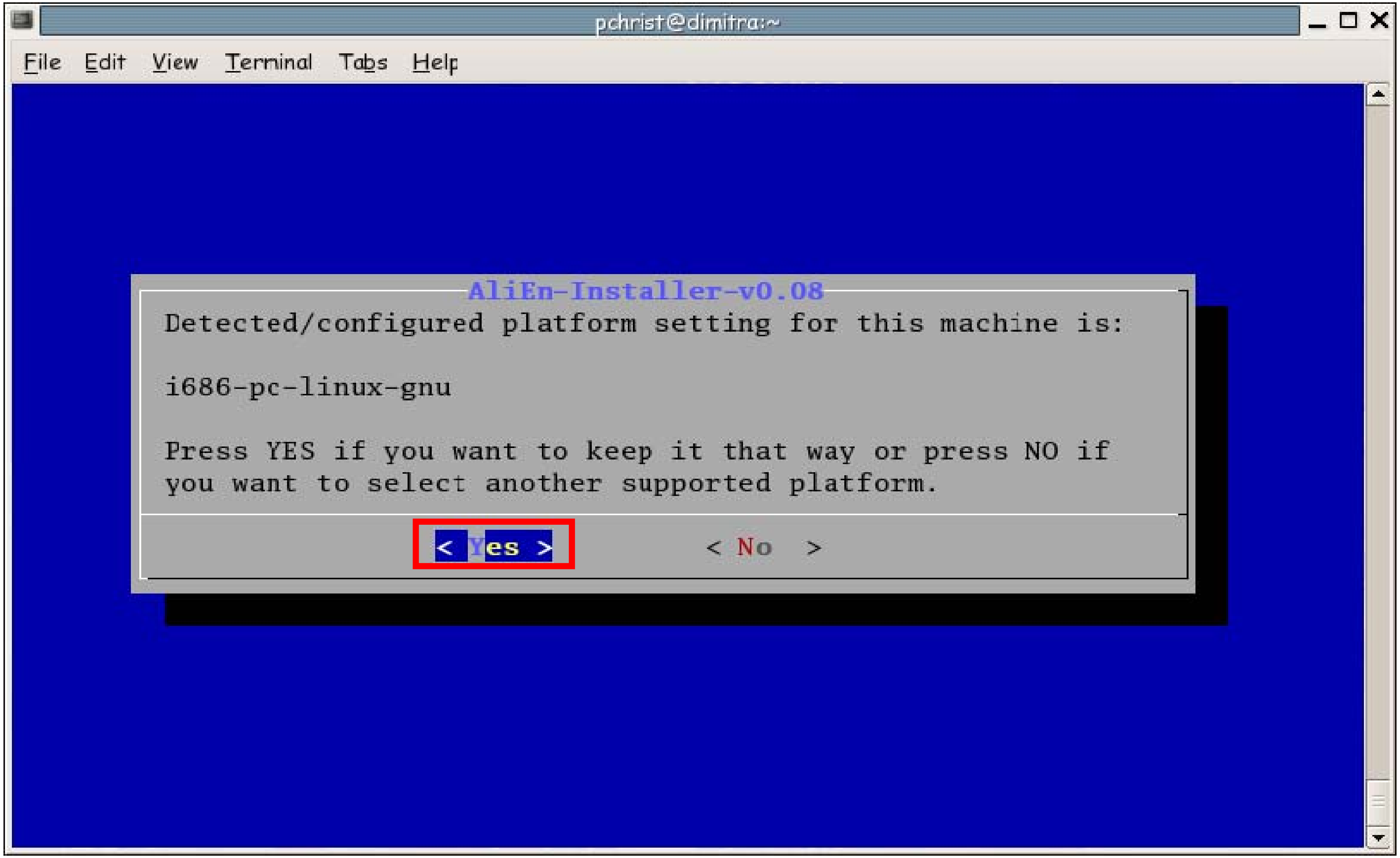
Installation – Workspace directory



```
Default workspace location for this installation is

/home/pchrist/.alien/cache

Press YES if you want to keep it that way or press NO if
you want to pick another location.
```

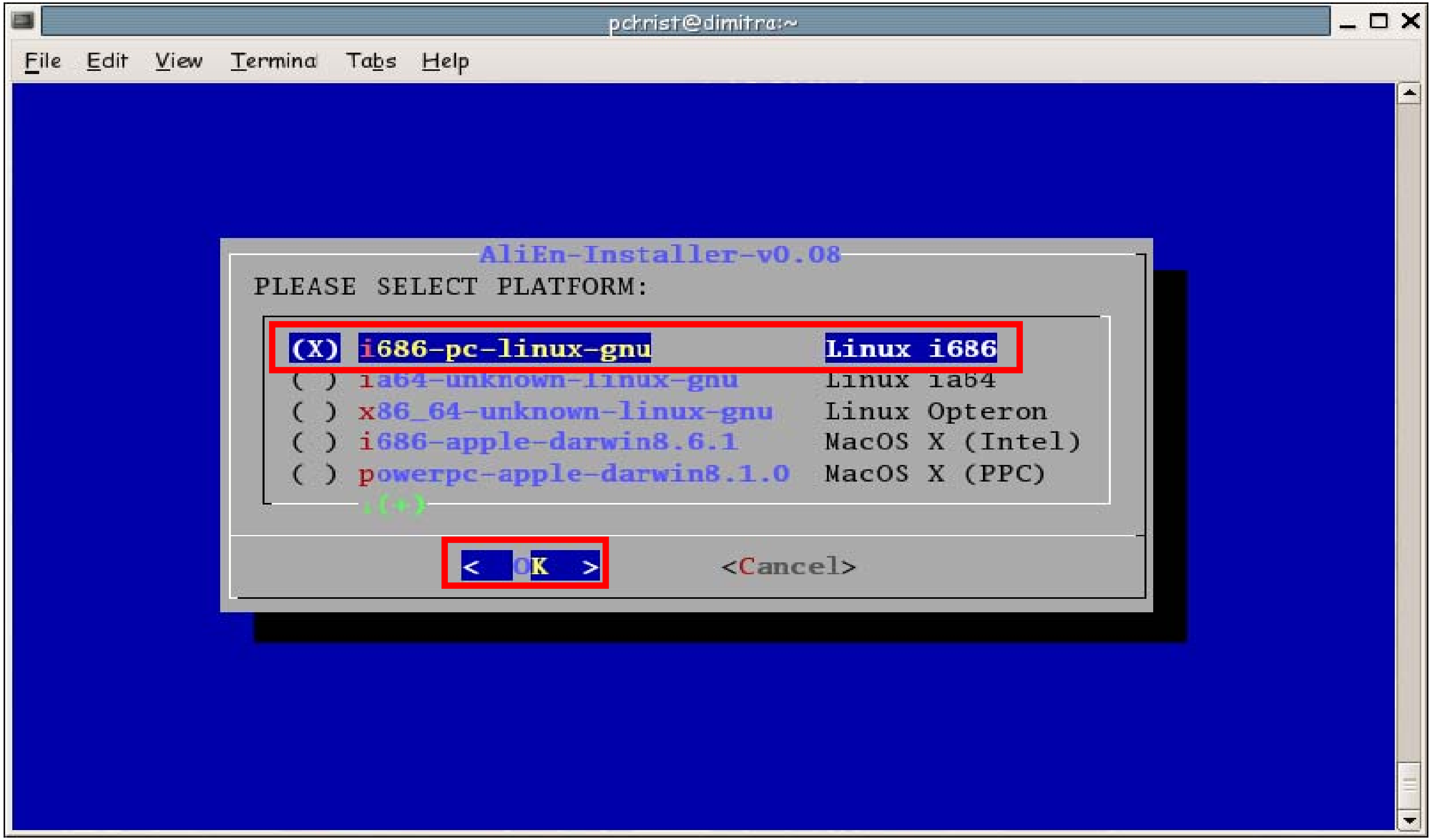
All the installation log files are stored in this directory!
In case of installation problems please send the error message along with the install.log file

Installation – Detecting the platftom



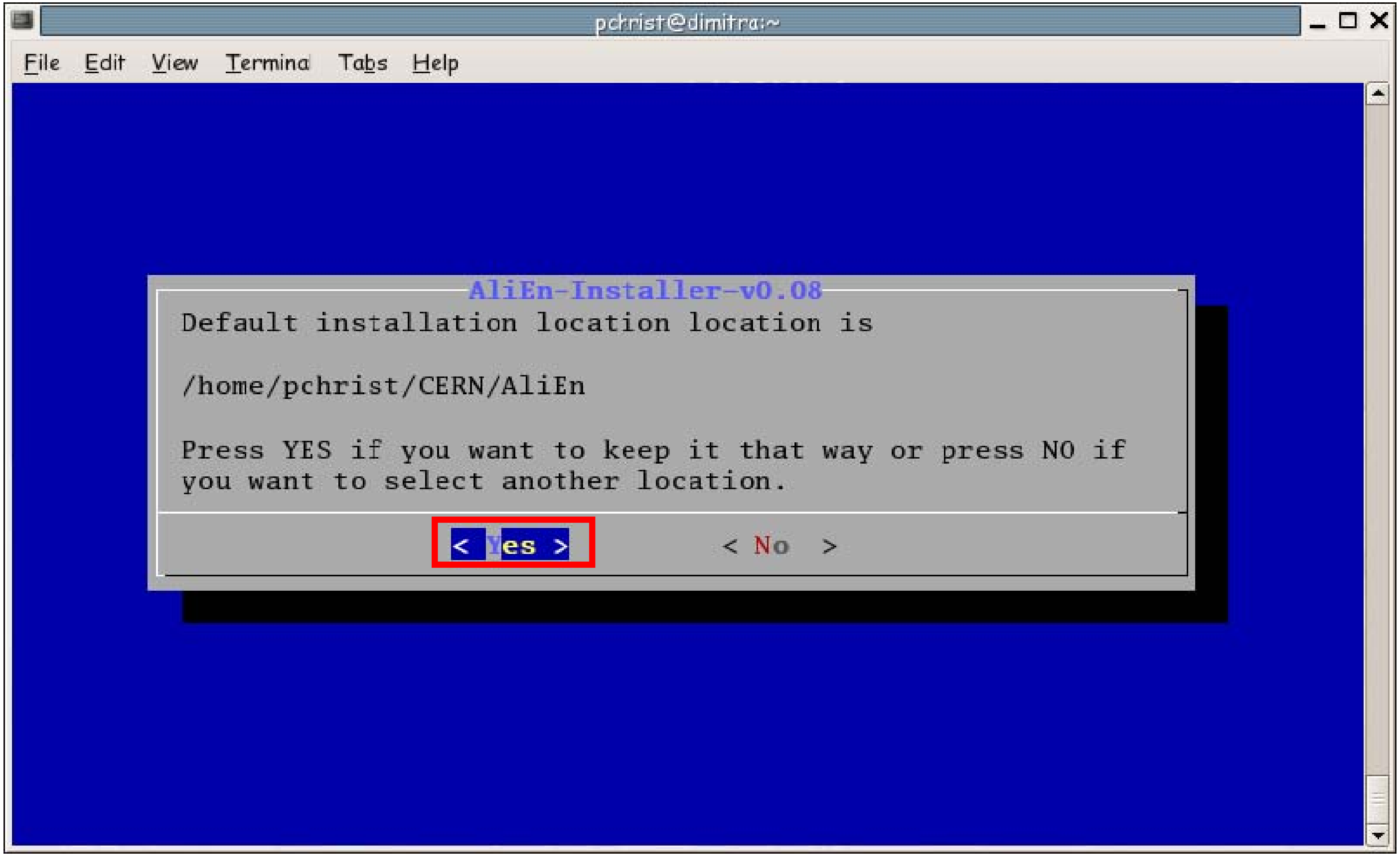
```
AliEn-Installer-v0.08
Detected/configured platform setting for this machine is:

i686-pc-linux-gnu

Press YES if you want to keep it that way or press NO if
you want to select another supported platform.
```

Installation – Supported platforms



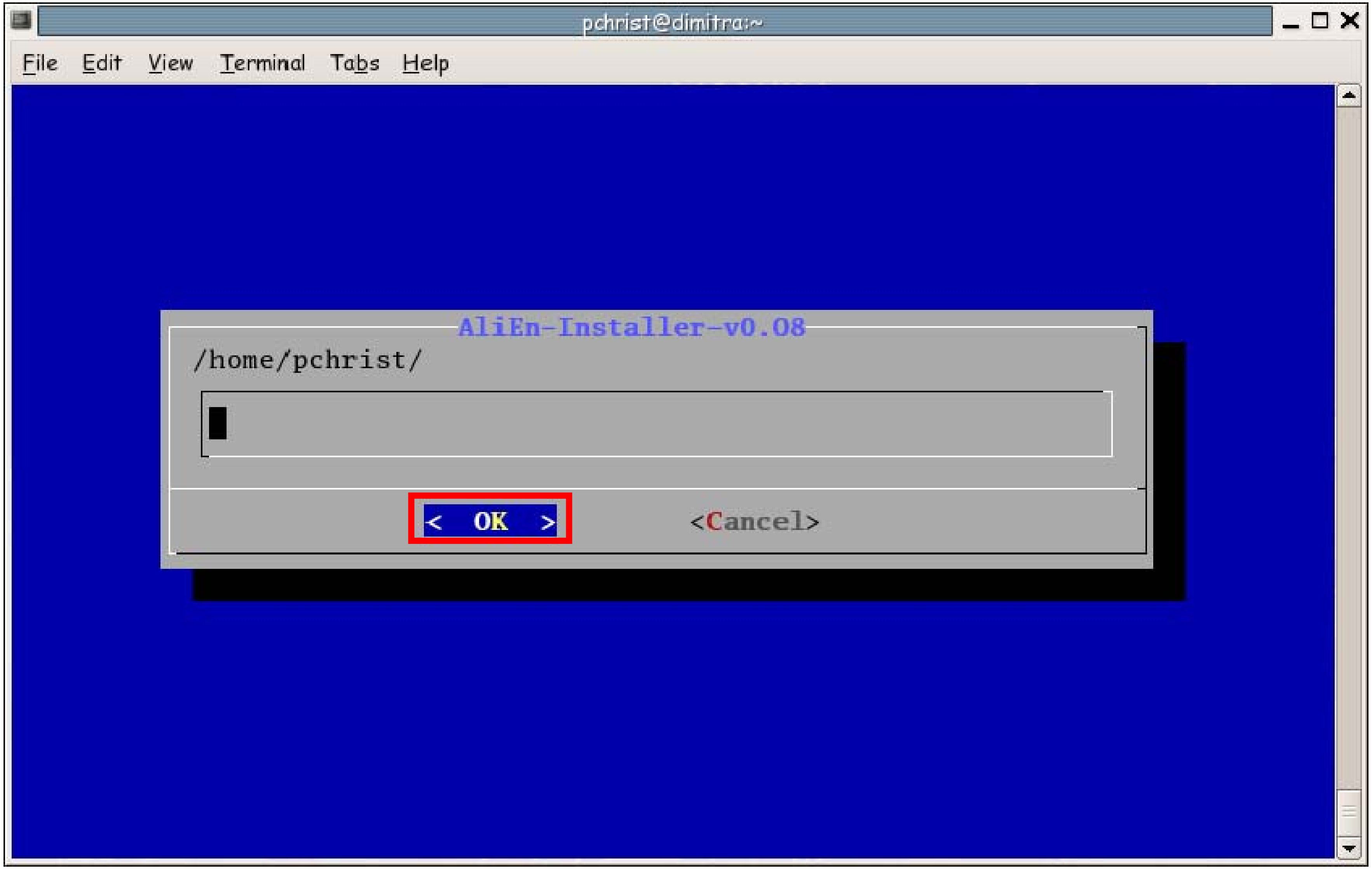
```
                AliEn-Installer-v0.08
 PLEASE SELECT PLATFORM:

   (X) i686-pc-linux-gnu            Linux i686
   ( ) ia64-unknown-linux-gnu       Linux ia64
   ( ) x86_64-unknown-linux-gnu     Linux Opteron
   ( ) i686-apple-darwin8.6.1       MacOS X (Intel)
   ( ) powerpc-apple-darwin8.1.0    MacOS X (PPC)
       ↓(+)

           <  OK  >           <Cancel>
```

Installation – Installation directory



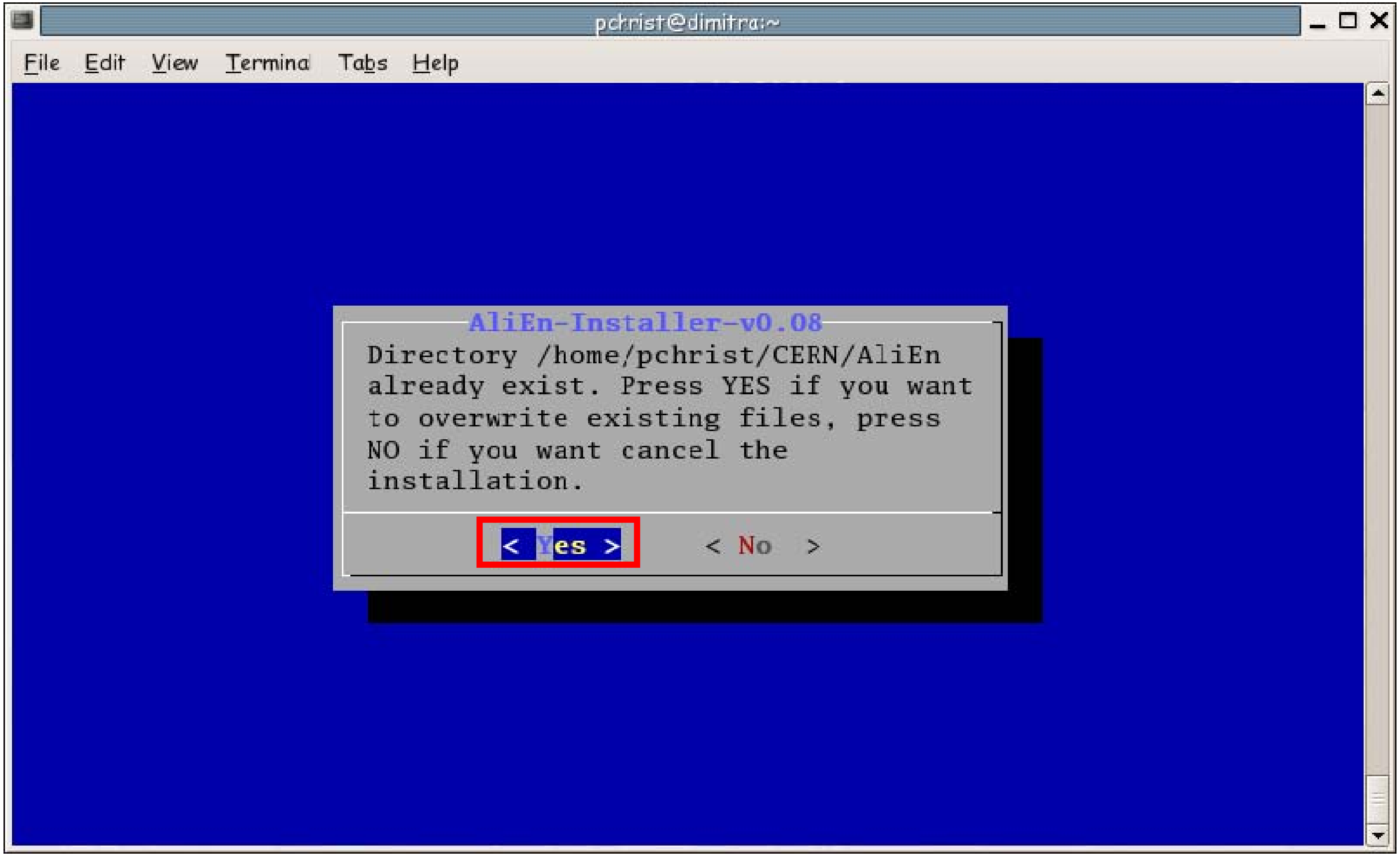
```
AliEn-Installer-v0.08
Default installation location location is

/home/pchrist/CERN/AliEn

Press YES if you want to keep it that way or press NO if
you want to select another location.

        < Yes >              < No  >
```

Installation – Selecting installation directory



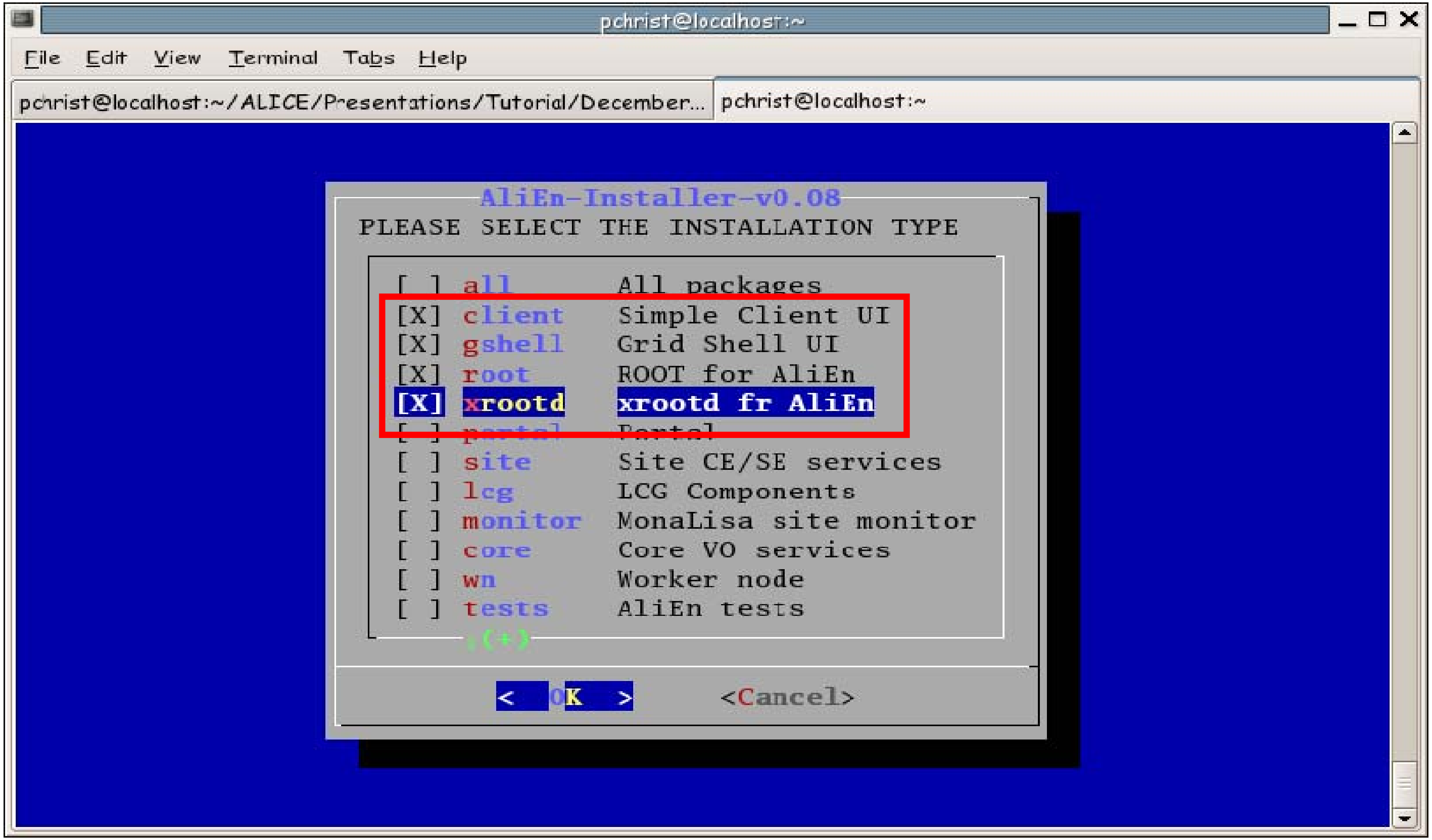
```
/home/pchrist/
```

Installation – Overwriting files



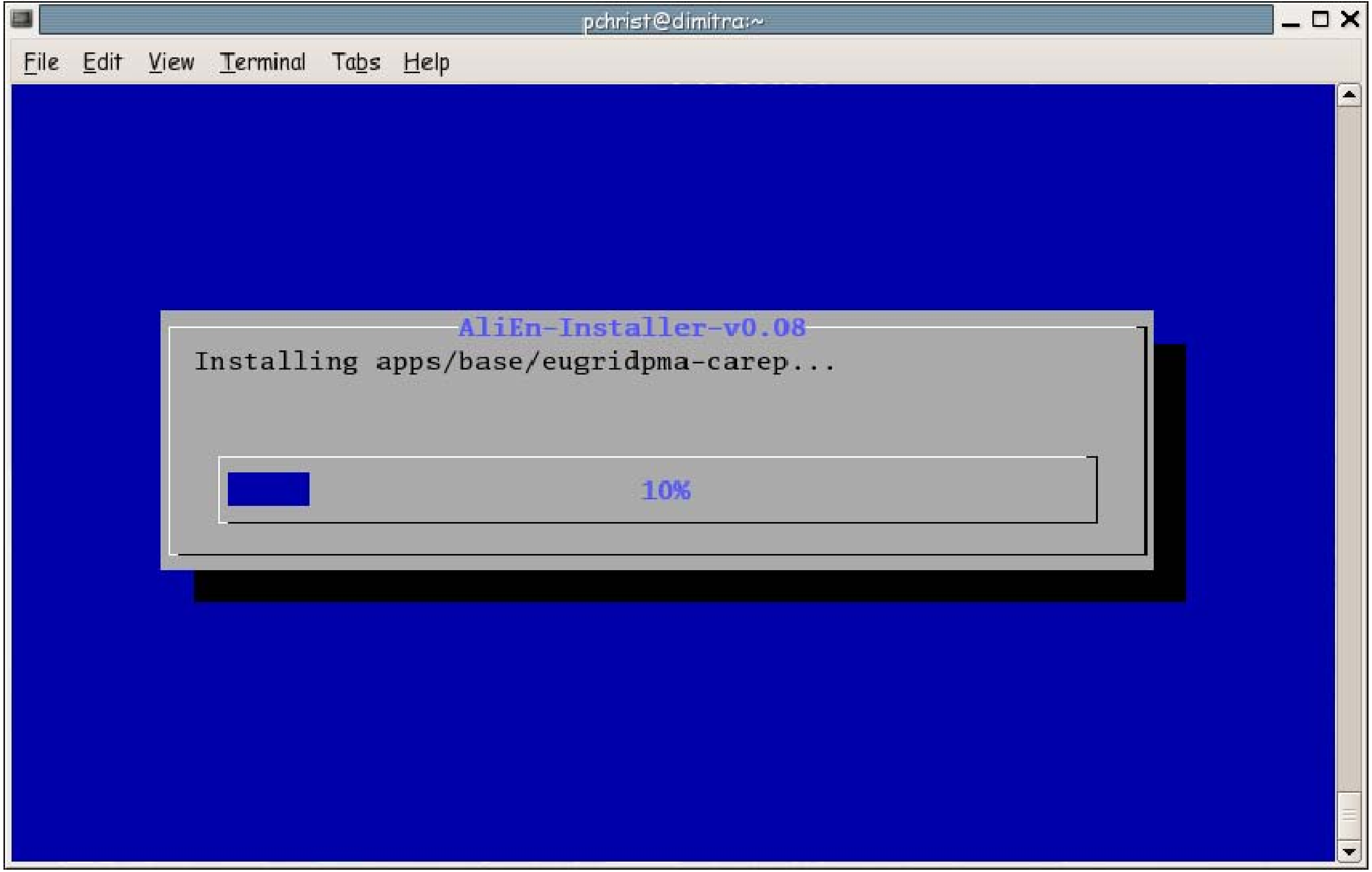
```
AliEn-Installer-v0.08
Directory /home/pchrist/CERN/AliEn
already exist. Press YES if you want
to overwrite existing files, press
NO if you want cancel the
installation.

        < Yes >        < No  >
```

Installation – Selecting packages


pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentations/Tutorial/December...
pchrist@localhost:~

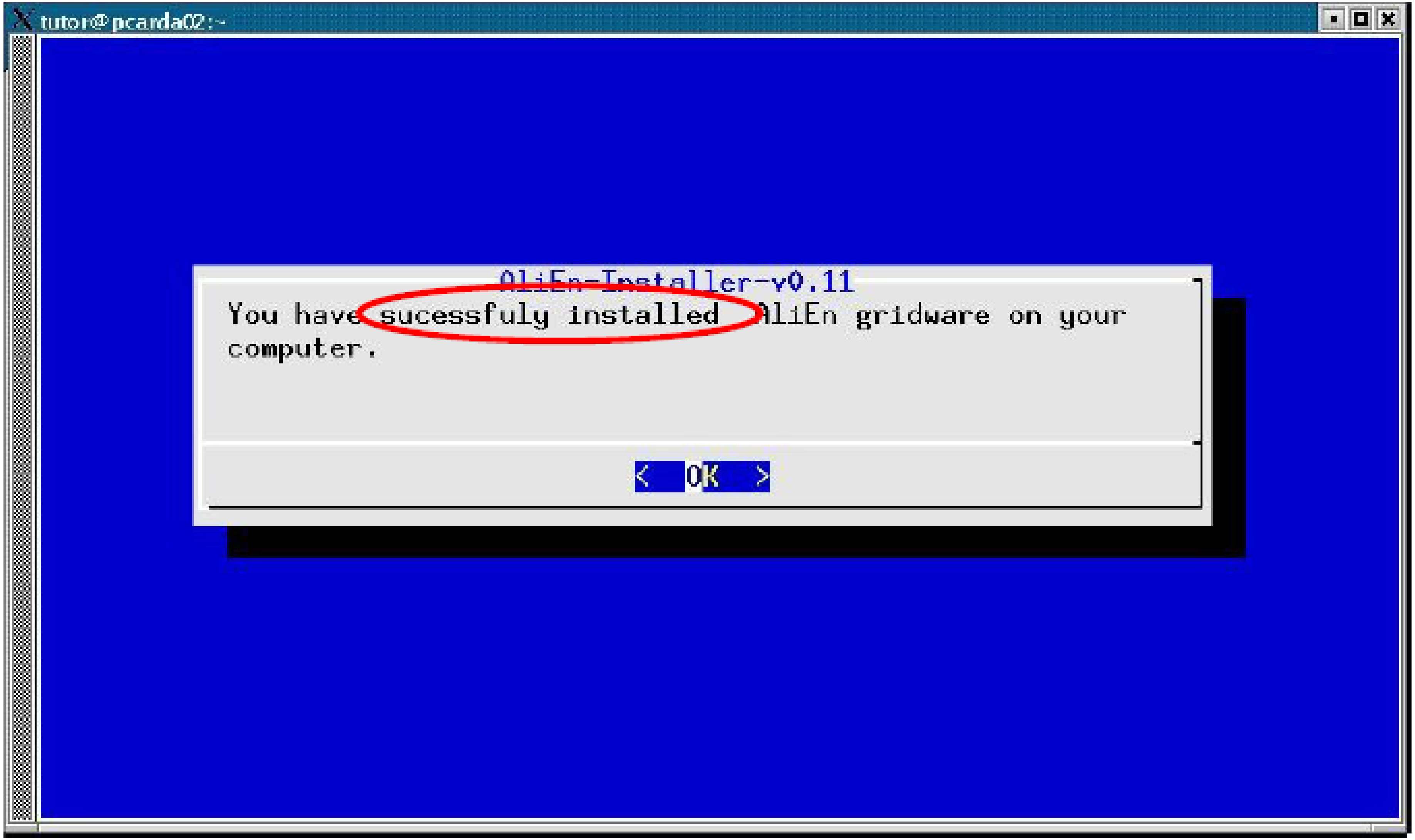
```
AliEn-Installer-v0.08
PLEASE SELECT THE INSTALLATION TYPE

  [ ] all      All packages
  [X] client   Simple Client UI
  [X] gshell   Grid Shell UI
  [X] root     ROOT for AliEn
  [X] xrootd   xrootd fr AliEn
  [ ] portal   Portal
  [ ] site     Site CE/SE services
  [ ] lcg      LCG Components
  [ ] monitor  MonaLisa site monitor
  [ ] core     Core VO services
  [ ] wn       Worker node
  [ ] tests    AliEn tests
      (+)

        <  OK  >        <Cancel>
```

Installation – Progress bar



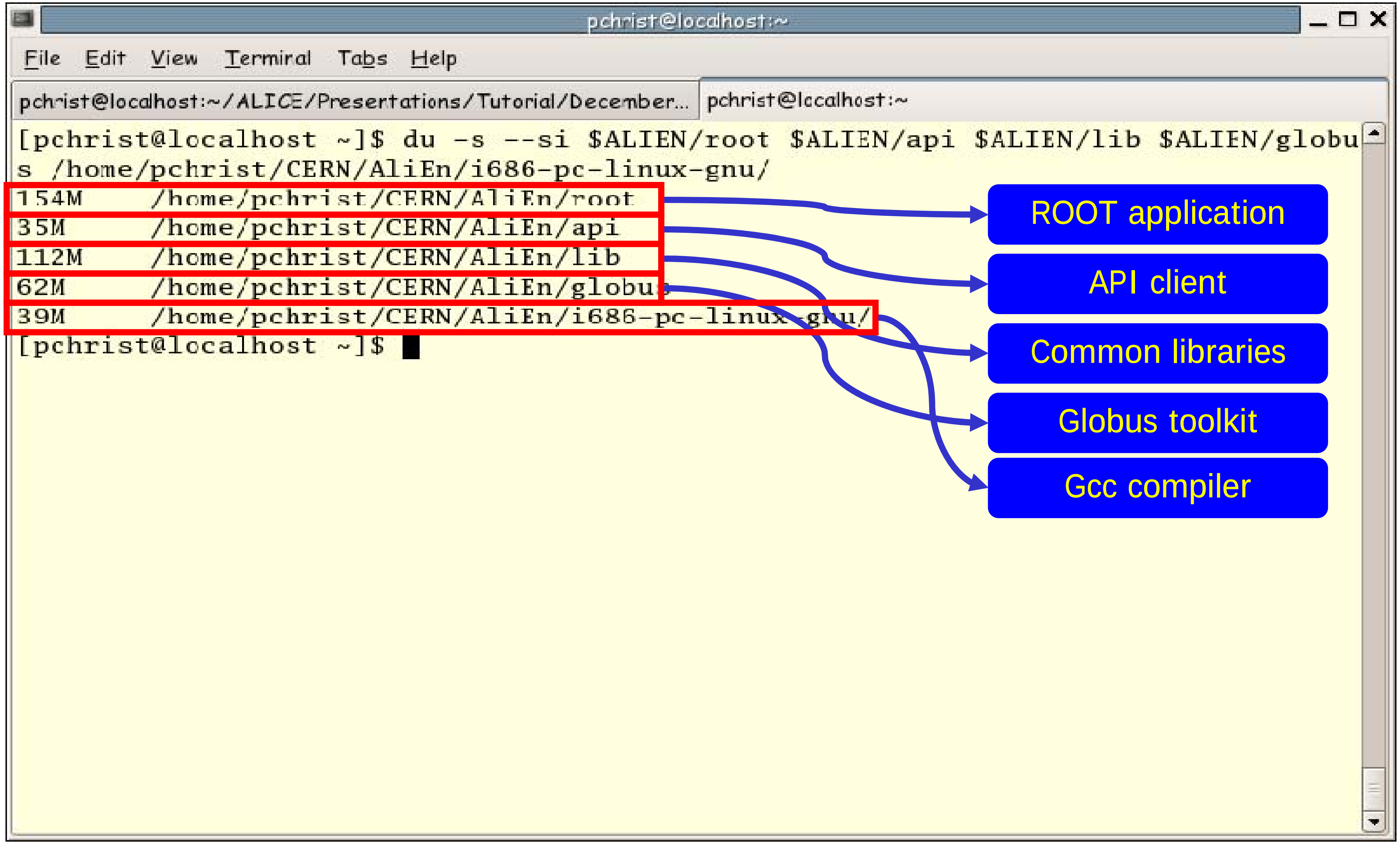
```
AliEn-Installer-v0.08
Installing apps/base/eugridpma-carep...

10%
```

Installation – Final window



```
You have sucessfuly installed AliEn gridware on your
computer.
```

Installation – Directory structure

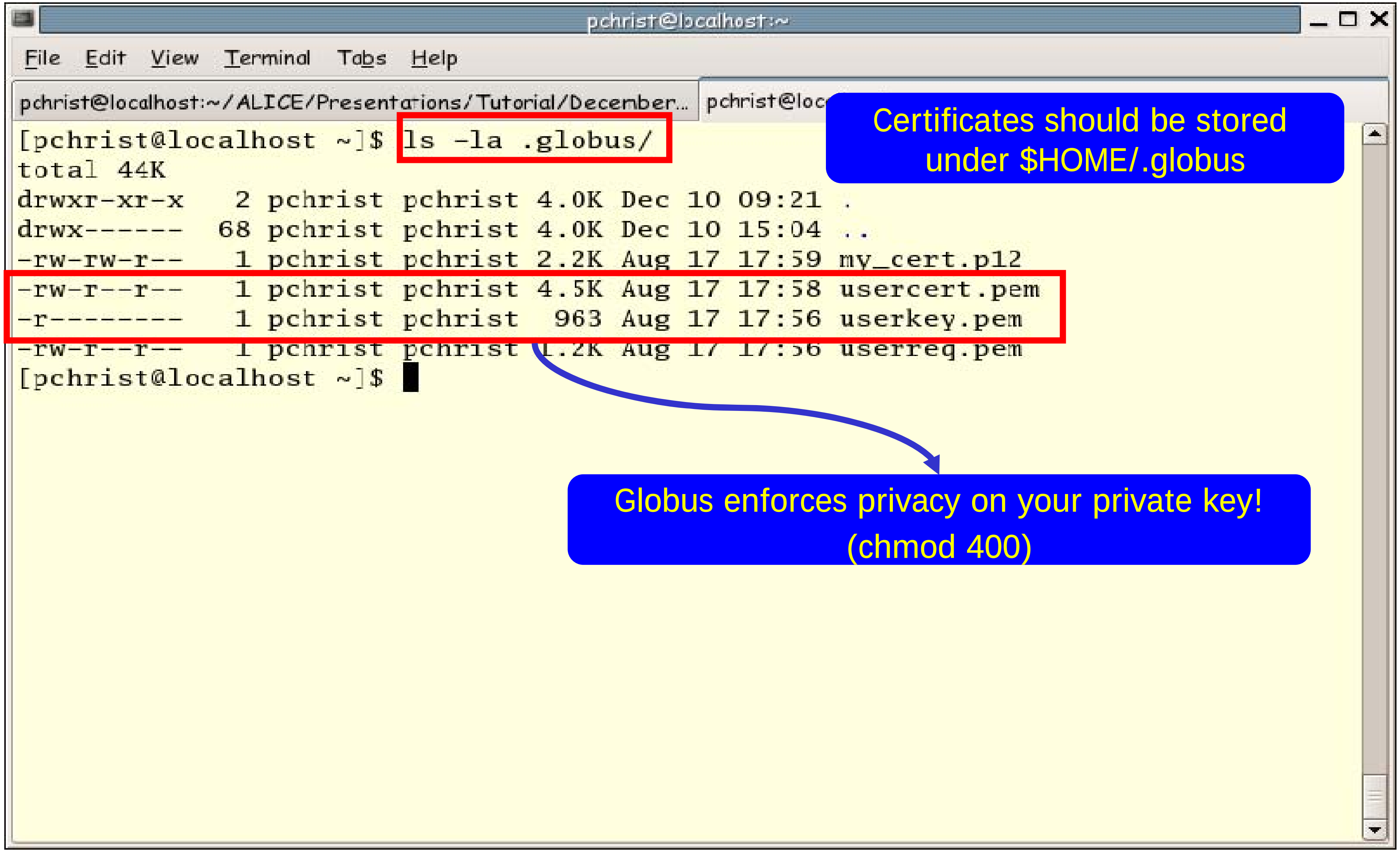

pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentations/Tutorial/December...
pchrist@localhost:~

```
[pchrist@localhost ~]$ du -s --si $ALIEN/root $ALIEN/api $ALIEN/lib $ALIEN/globu
s /home/pchrist/CERN/AliEn/i686-pc-linux-gnu/
154M    /home/pchrist/CERN/AliEn/root
35M     /home/pchrist/CERN/AliEn/api
112M    /home/pchrist/CERN/AliEn/lib
62M     /home/pchrist/CERN/AliEn/globus
39M     /home/pchrist/CERN/AliEn/i686-pc-linux-gnu/
[pchrist@localhost ~]$
```

Installation – Try it out

- Download the alien installer from **http://alien.cern.ch**.
- Make the file executable.
- Run the installer.
- Select v2-12
- Platform should be i686.
- Select installation directory.
- Select the following packages:
  - Client
  - gShell
  - ROOT
  - xrootd

Authentication – Preparing the certificates

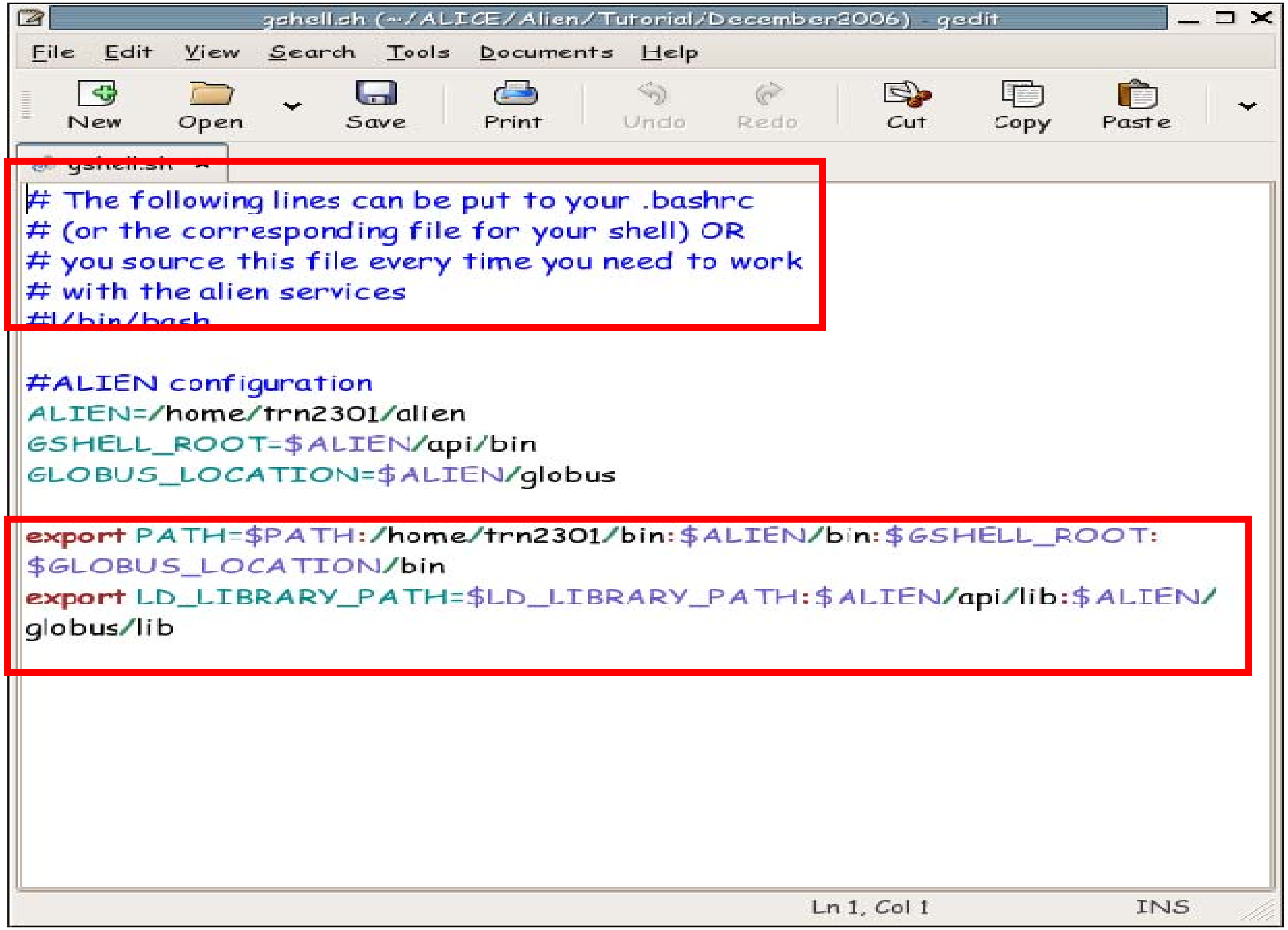

pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentations/Tutorial/December...
pchrist@loc

```
[pchrist@localhost ~]$ ls -la .globus/
total 44K
drwxr-xr-x   2 pchrist pchrist 4.0K Dec 10 09:21 .
drwx------  68 pchrist pchrist 4.0K Dec 10 15:04 ..
-rw-rw-r--   1 pchrist pchrist 2.2K Aug 17 17:59 my_cert.p12
-rw-r--r--   1 pchrist pchrist 4.5K Aug 17 17:58 usercert.pem
-r--------   1 pchrist pchrist  963 Aug 17 17:56 userkey.pem
-rw-r--r--   1 pchrist pchrist 1.2K Aug 17 17:56 userreq.pem
[pchrist@localhost ~]$
```

Certificates should be stored under $HOME/.globus

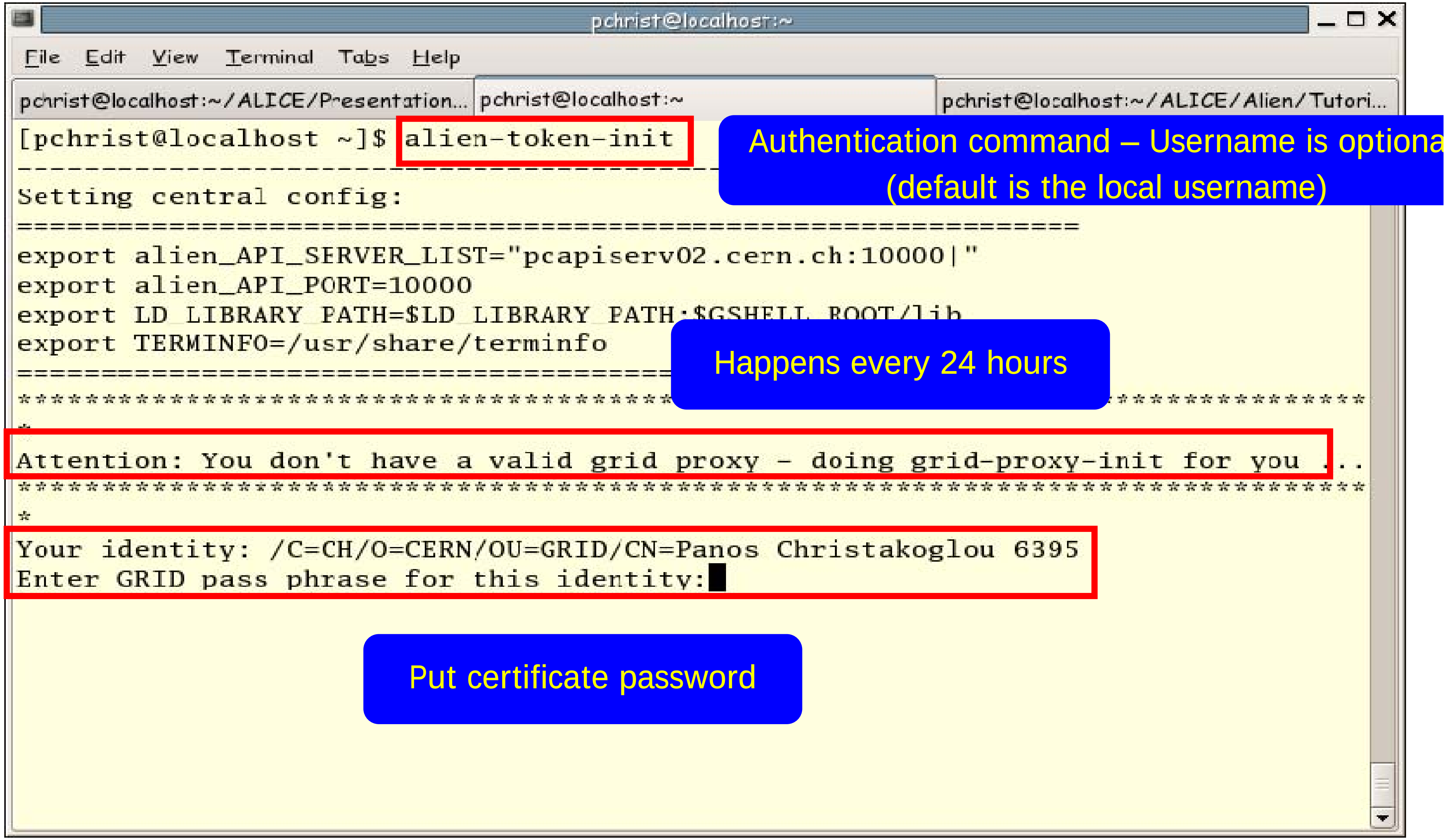
Globus enforces privacy on your private key! (chmod 400)

Authentication – Changes in env. variables


gshell.sh (~/ALICE/Alien/Tutorial/December2006) - gedit
File Edit View Search Tools Documents Help
New
Open
Save
Print
Undo
Redo
Cut
Copy
Paste
gshell.sh
Ln 1, Col 1
INS

```
# The following lines can be put to your .bashrc
# (or the corresponding file for your shell) OR
# you source this file every time you need to work
# with the alien services
#!/bin/bash

#ALIEN configuration
ALIEN=/home/trn2301/alien
GSHELL_ROOT=$ALIEN/api/bin
GLOBUS_LOCATION=$ALIEN/globus

export PATH=$PATH:/home/trn2301/bin:$ALIEN/bin:$GSHELL_ROOT:
$GLOBUS_LOCATION/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ALIEN/api/lib:$ALIEN/
globus/lib
```

Authentication – Getting a GRID proxy


pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

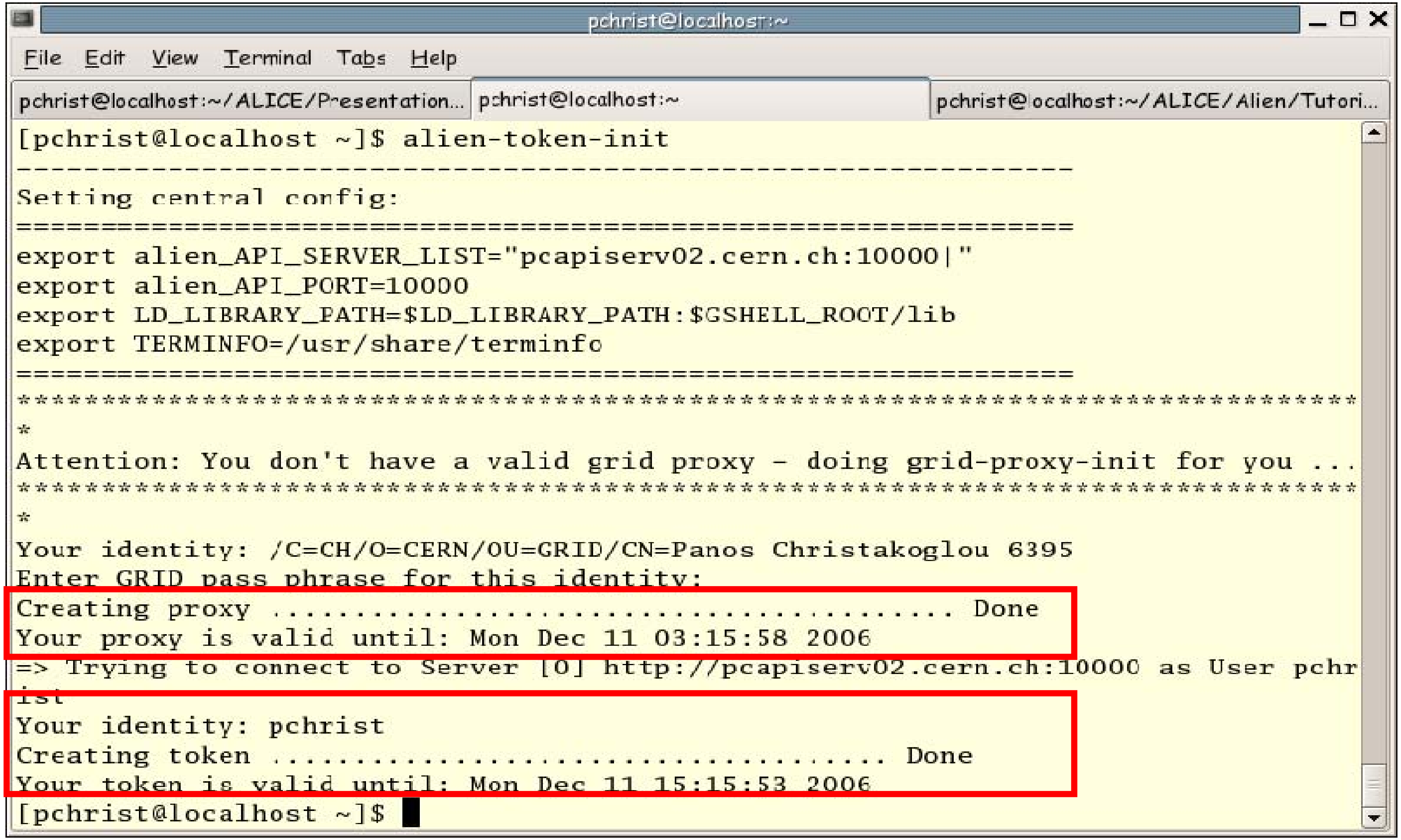
```
[pchrist@localhost ~]$ alien-token-init
------------------------------------------
Setting central config:
==============================================================
export alien_API_SERVER_LIST="pcapiserv02.cern.ch:10000|"
export alien_API_PORT=10000
export LD LIBRARY PATH=$LD LIBRARY PATH:$GSHELL ROOT/lib
export TERMINFO=/usr/share/terminfo
=========================================
****************************************               ****************
*
Attention: You don't have a valid grid proxy - doing grid-proxy-init for you ...
********************************************************************************
*
Your identity: /C=CH/O=CERN/OU=GRID/CN=Panos Christakoglou 6395
Enter GRID pass phrase for this identity:
```

Authentication command – Username is optiona
(default is the local username)

Happens every 24 hours

Put certificate password

Authentication – Getting an AliEn token


pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
[pchrist@localhost ~]$ alien-token-init
----------------------------------------------------------------
Setting central config:
================================================================
export alien_API_SERVER_LIST="pcapiserv02.cern.ch:10000|"
export alien_API_PORT=10000
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$GSHELL_ROOT/lib
export TERMINFO=/usr/share/terminfo
================================================================
********************************************************************************
*
Attention: You don't have a valid grid proxy - doing grid-proxy-init for you ...
********************************************************************************
*
Your identity: /C=CH/O=CERN/OU=GRID/CN=Panos Christakoglou 6395
Enter GRID pass phrase for this identity:
Creating proxy ........................................ Done
Your proxy is valid until: Mon Dec 11 03:15:58 2006
=> Trying to connect to Server [0] http://pcapiserv02.cern.ch:10000 as User pchr
ist
Your identity: pchrist
Creating token .................................... Done
Your token is valid until: Mon Dec 11 15:15:53 2006
[pchrist@localhost ~]$
```

Authentication – Authentication problems I

- Globus related:
  - Permissions on $HOME/.globus/userkey.pem are not private to the user – chmod 400 userkey.pem
  - Your certificate authority is exotic and not known to the server.
  - Your certificate has expired.
  - Clock skew:
    - Your local computer time is in the future with respect to the server's time.
    - Your local computer time is more in the past than the certificate life time.

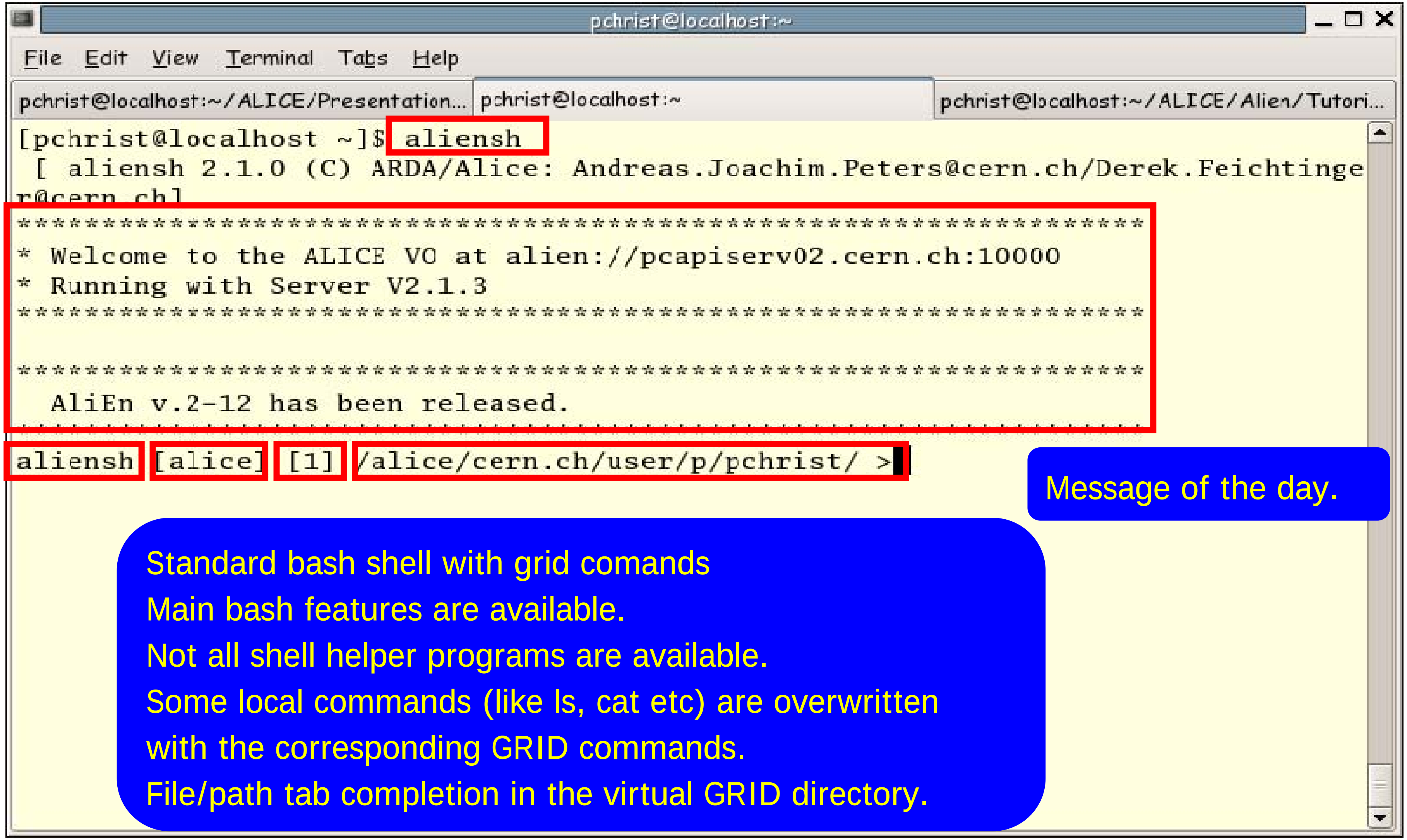
Authentication – Authentication problems II

- alien-token-init related:
  - You have not gone through all 5 steps of the AliEn user registration.
  - You have not given the AliEn user name as an argument to the token-init command and your local user name is not identical to the AliEn user name.
  - The script wants to bootstrap the installation but you don't have write permissions on the installation path – Avoid bootstrapping by setting the GSHELL_ROOT environment variable.

Authentication – Try it out

- Upload your certificates to your machines:
  - Store them under e.g: /afs/cern.ch/user/t/trn2301/.globus/
- Get the gshell.sh file from the agenda and place it under e.g. /home/trn2301/
- Open it and change it accordingly.
- Get a valid grid proxy.
- Get a valid alien token.
- Check the information of your proxy/token by typing:
  - grid-proxy-info
  - alien-token-info

Shell (1) – Accessing the shell

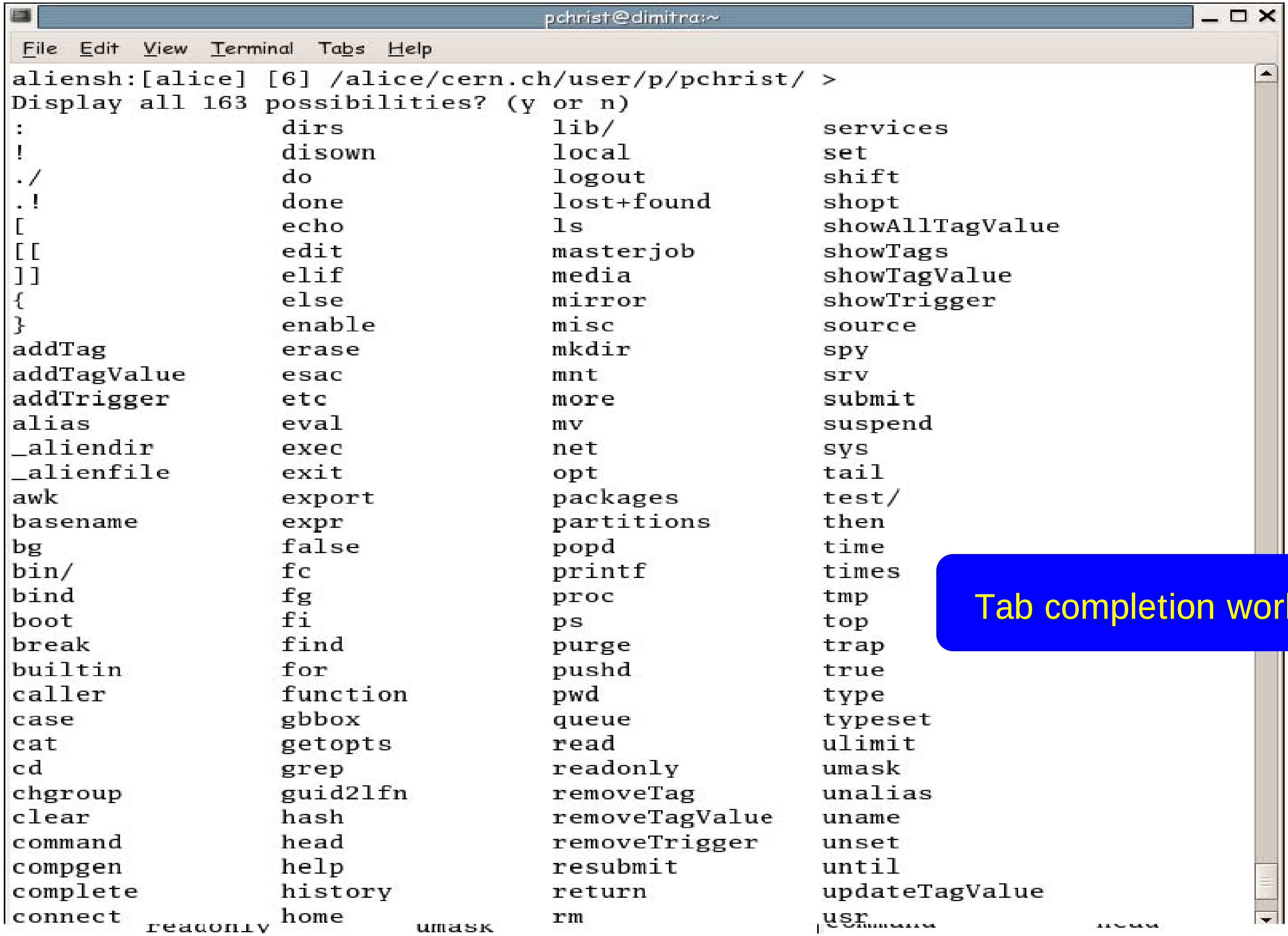

pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
[pchrist@localhost ~]$ aliensh
 [ aliensh 2.1.0 (C) ARDA/Alice: Andreas.Joachim.Peters@cern.ch/Derek.Feichtinge
r@cern.ch]
********************************************************************
* Welcome to the ALICE VO at alien://pcapiserv02.cern.ch:10000
* Running with Server V2.1.3
********************************************************************

********************************************************************
  AliEn v.2-12 has been released.
********************************************************************
aliensh [alice] [1] /alice/cern.ch/user/p/pchrist/ >
```

Message of the day.

Standard bash shell with grid comands
Main bash features are available.
Not all shell helper programs are available.
Some local commands (like ls, cat etc) are overwritten with the corresponding GRID commands.
File/path tab completion in the virtual GRID directory.

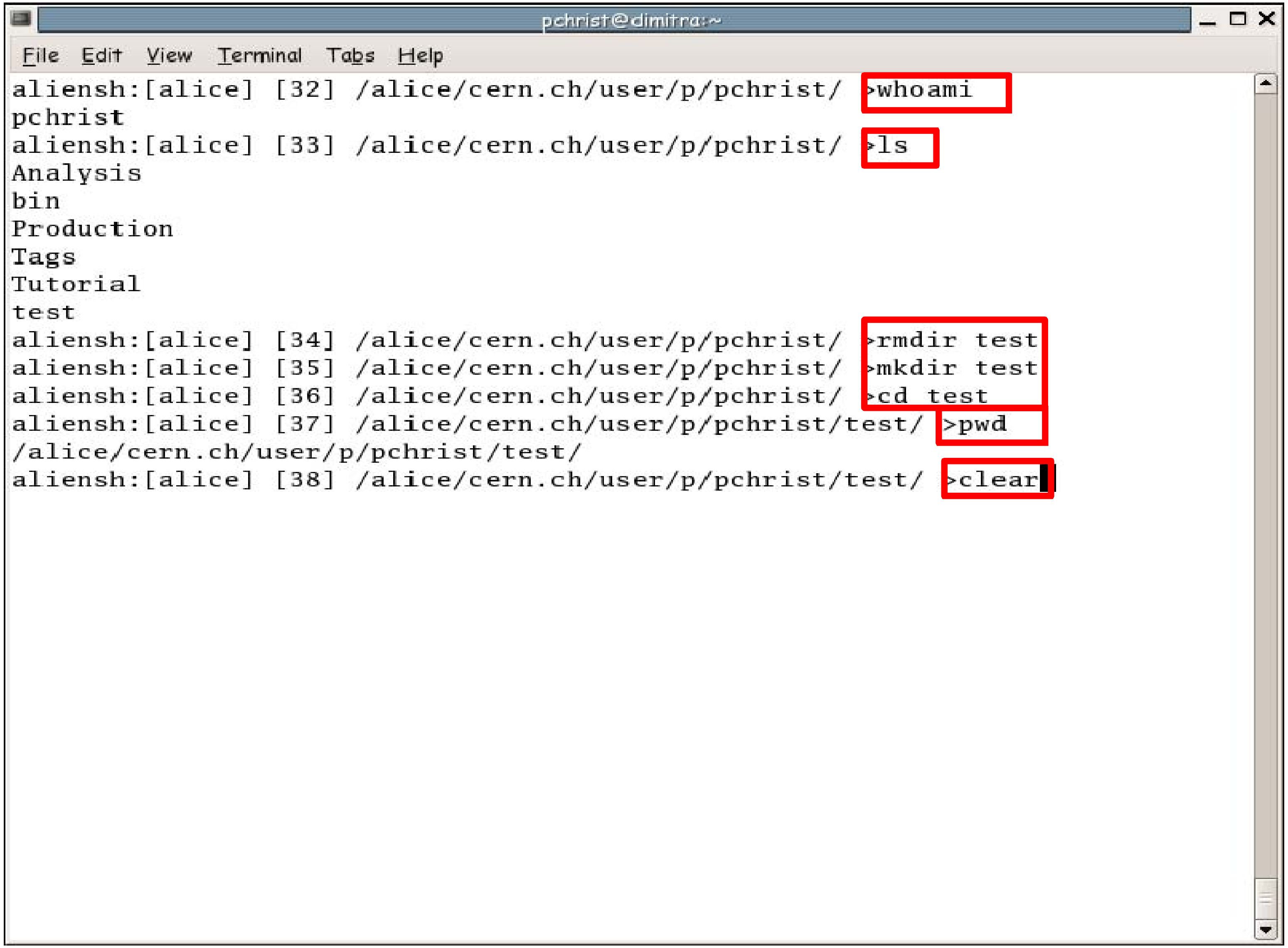
Shell (1) – Basic commands I



```
aliensh:[alice] [6] /alice/cern.ch/user/p/pchrist/ >
Display all 163 possibilities? (y or n)
:                dirs             lib/             services
!                disown           local            set
./               do               logout           shift
.!               done             lost+found       shopt
[                echo             ls               showAllTagValue
[[               edit             masterjob        showTags
]]               elif             media            showTagValue
{                else             mirror           showTrigger
}                enable           misc             source
addTag           erase            mkdir            spy
addTagValue      esac             mnt              srv
addTrigger       etc              more             submit
alias            eval             mv               suspend
_aliendir        exec             net              sys
_alienfile       exit             opt              tail
awk              export           packages         test/
basename         expr             partitions       then
bg               false            popd             time
bin/             fc               printf           times
bind             fg               proc             tmp
boot             fi               ps               top
break            find             purge            trap
builtin          for              pushd            true
caller           function         pwd              type
case             gbbox            queue            typeset
cat              getopts          read             ulimit
cd               grep             readonly         umask
chgroup          guid2lfn         removeTag        unalias
clear            hash             removeTagValue   uname
command          head             removeTrigger    unset
compgen          help             resubmit         until
complete         history          return           updateTagValue
connect          home             rm               usr
```

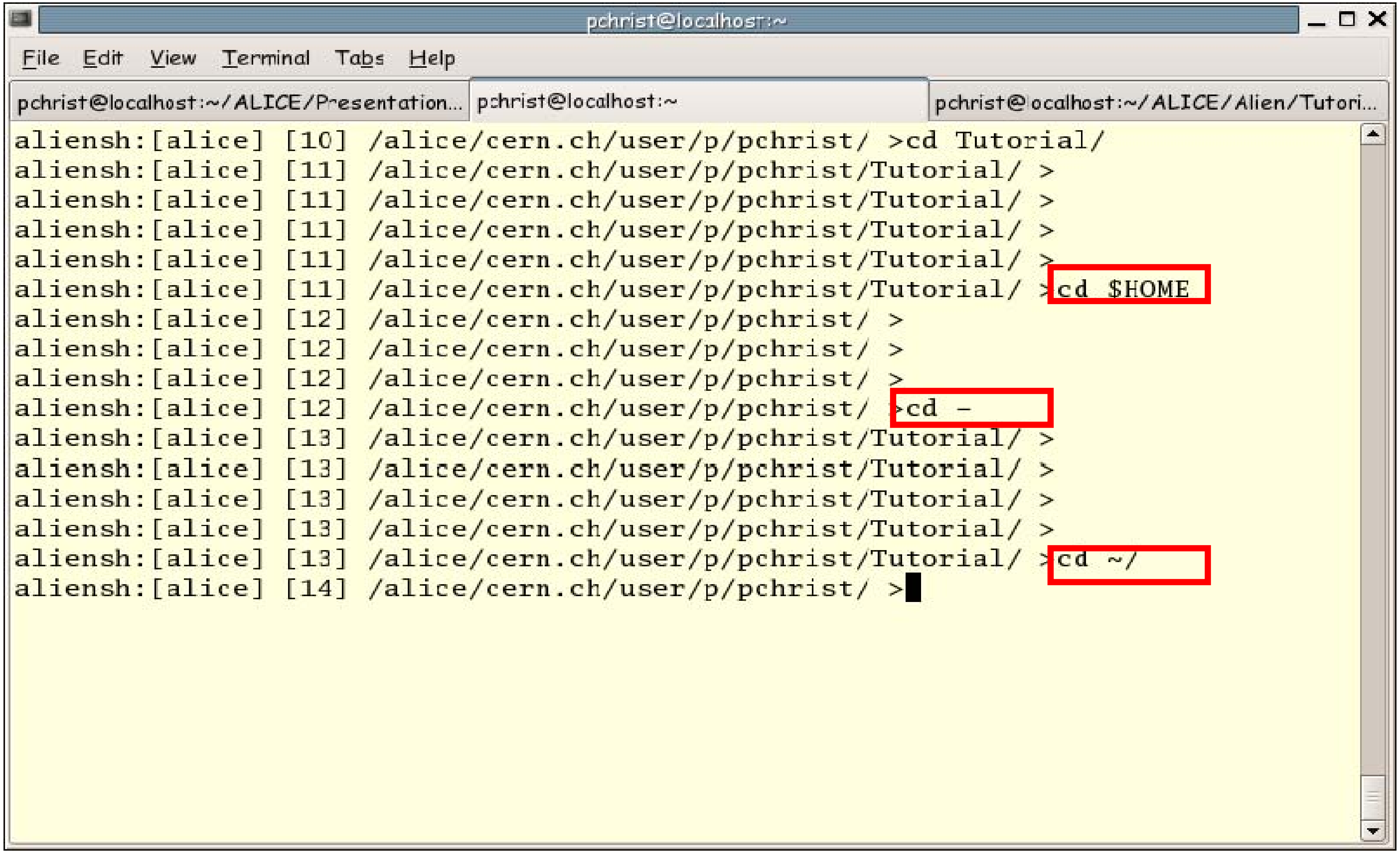
Tab completion working!!!

Shell (1) – Basic commands II



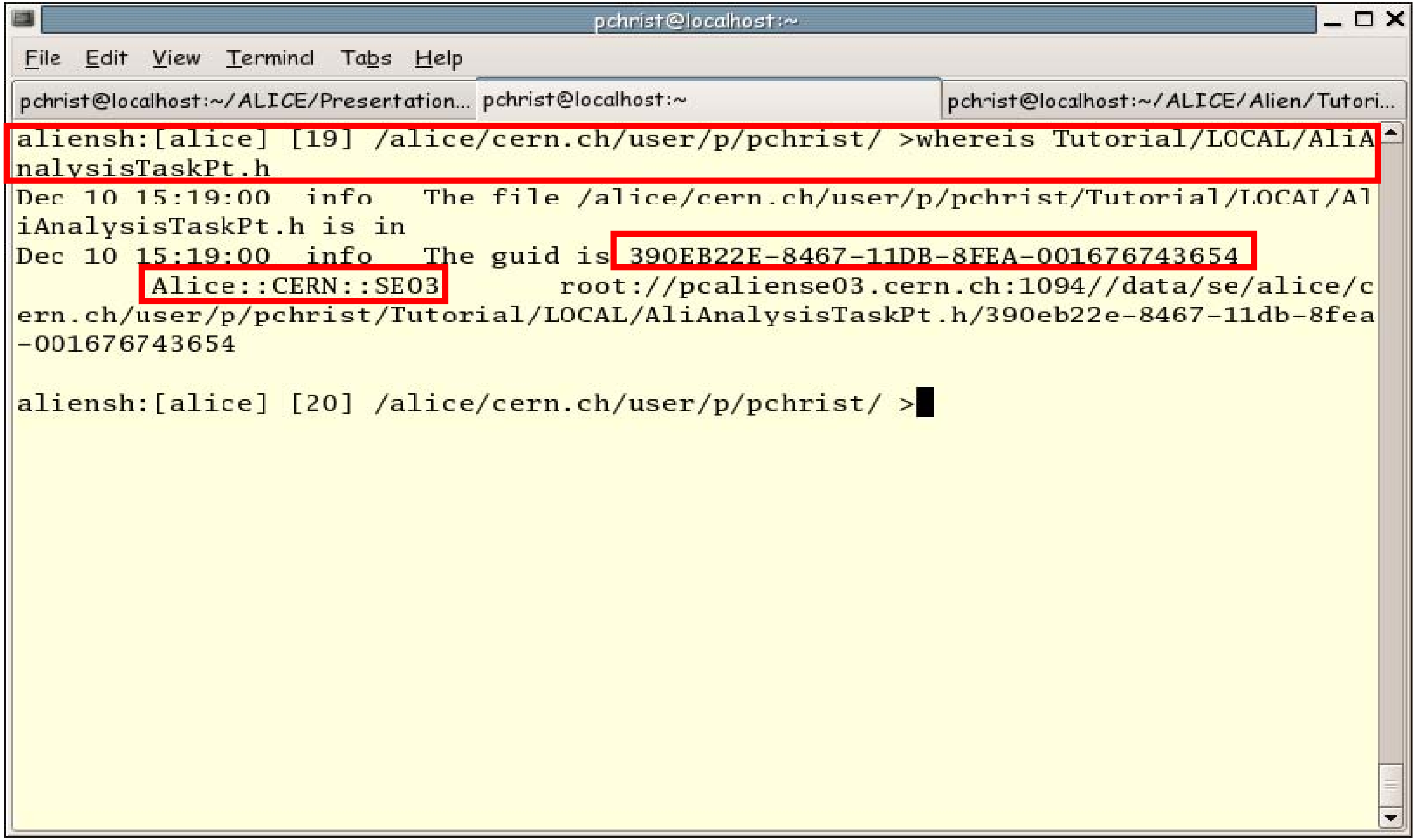
```
aliensh:[alice] [32] /alice/cern.ch/user/p/pchrist/ >whoami
pchrist
aliensh:[alice] [33] /alice/cern.ch/user/p/pchrist/ >ls
Analysis
bin
Production
Tags
Tutorial
test
aliensh:[alice] [34] /alice/cern.ch/user/p/pchrist/ >rmdir test
aliensh:[alice] [35] /alice/cern.ch/user/p/pchrist/ >mkdir test
aliensh:[alice] [36] /alice/cern.ch/user/p/pchrist/ >cd test
aliensh:[alice] [37] /alice/cern.ch/user/p/pchrist/test/ >pwd
/alice/cern.ch/user/p/pchrist/test/
aliensh:[alice] [38] /alice/cern.ch/user/p/pchrist/test/ >clear
```

Shell (1) – Basic commands III


pchrist@localhost:~
File
Edit
View
Terminal
Tabs
Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
aliensh:[alice] [10] /alice/cern.ch/user/p/pchrist/ >cd Tutorial/
aliensh:[alice] [11] /alice/cern.ch/user/p/pchrist/Tutorial/ >
aliensh:[alice] [11] /alice/cern.ch/user/p/pchrist/Tutorial/ >
aliensh:[alice] [11] /alice/cern.ch/user/p/pchrist/Tutorial/ >
aliensh:[alice] [11] /alice/cern.ch/user/p/pchrist/Tutorial/ >
aliensh:[alice] [11] /alice/cern.ch/user/p/pchrist/Tutorial/ >cd $HOME
aliensh:[alice] [12] /alice/cern.ch/user/p/pchrist/ >
aliensh:[alice] [12] /alice/cern.ch/user/p/pchrist/ >
aliensh:[alice] [12] /alice/cern.ch/user/p/pchrist/ >
aliensh:[alice] [12] /alice/cern.ch/user/p/pchrist/ >cd -
aliensh:[alice] [13] /alice/cern.ch/user/p/pchrist/Tutorial/ >
aliensh:[alice] [13] /alice/cern.ch/user/p/pchrist/Tutorial/ >
aliensh:[alice] [13] /alice/cern.ch/user/p/pchrist/Tutorial/ >
aliensh:[alice] [13] /alice/cern.ch/user/p/pchrist/Tutorial/ >
aliensh:[alice] [13] /alice/cern.ch/user/p/pchrist/Tutorial/ >cd ~/
aliensh:[alice] [14] /alice/cern.ch/user/p/pchrist/ >
```

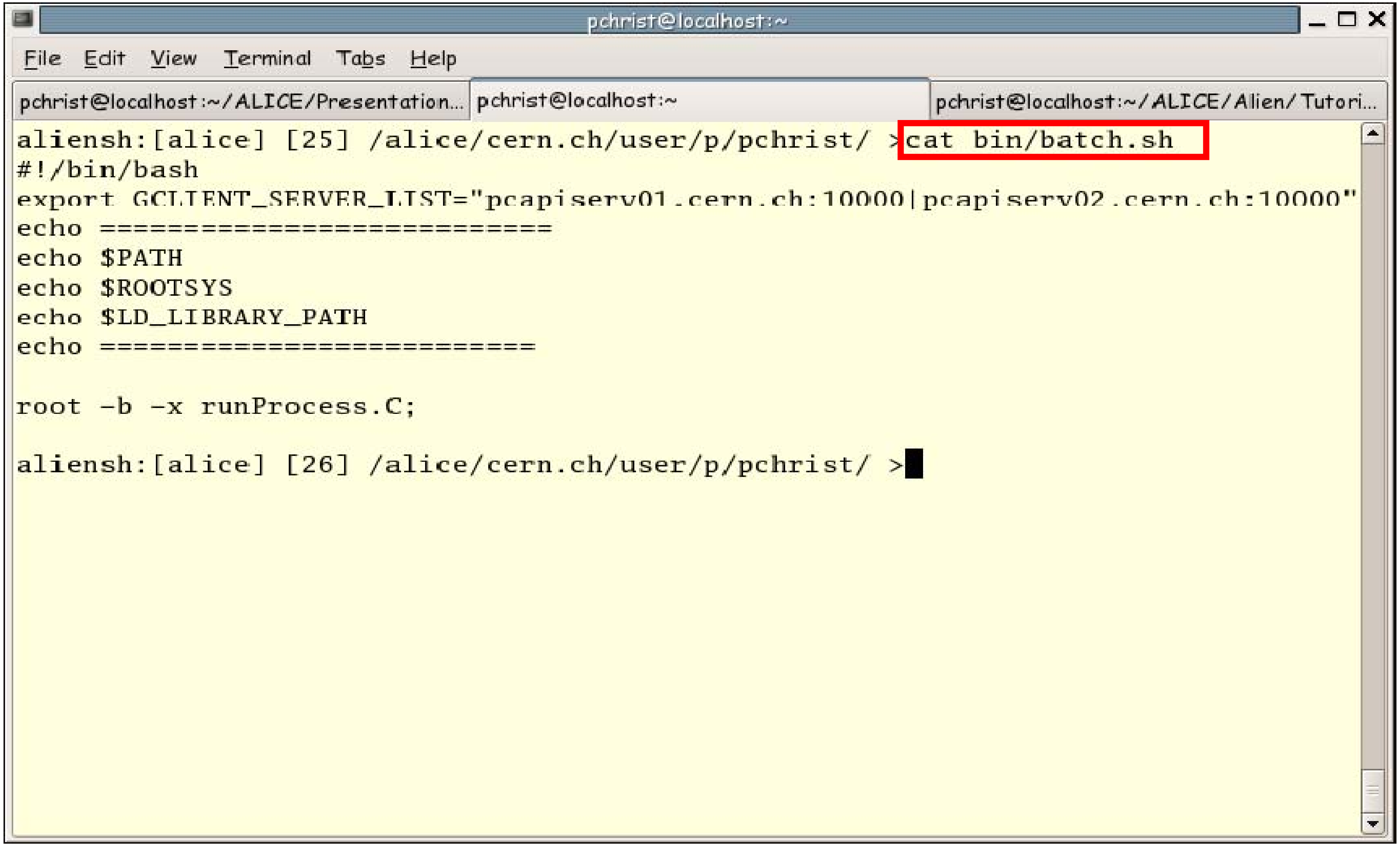
Shell (1) – whereis command



pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

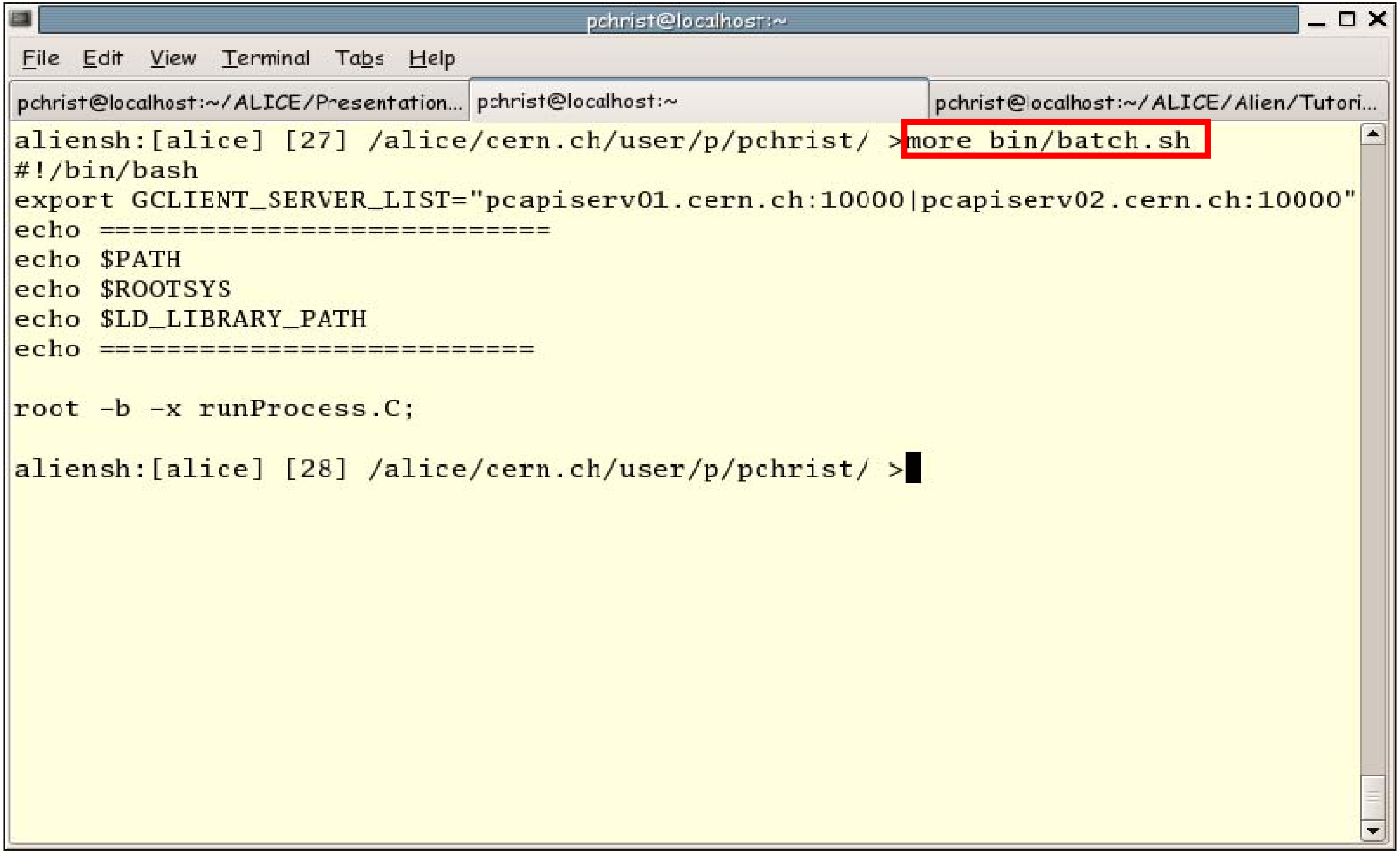
```
aliensh:[alice] [19] /alice/cern.ch/user/p/pchrist/ >whereis Tutorial/LOCAL/AliA
nalysisTaskPt.h
Dec 10 15:19:00  info   The file /alice/cern.ch/user/p/pchrist/Tutorial/LOCAL/Al
iAnalysisTaskPt.h is in
Dec 10 15:19:00  info   The guid is 390EB22E-8467-11DB-8FEA-001676743654
        Alice::CERN::SE03       root://pcaliense03.cern.ch:1094//data/se/alice/c
ern.ch/user/p/pchrist/Tutorial/LOCAL/AliAnalysisTaskPt.h/390eb22e-8467-11db-8fea
-001676743654

aliensh:[alice] [20] /alice/cern.ch/user/p/pchrist/ >
```

Shell (1) – Viewing the files I


pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
aliensh:[alice] [25] /alice/cern.ch/user/p/pchrist/ >cat bin/batch.sh
#!/bin/bash
export GCLIENT_SERVER_LIST="pcapiserv01.cern.ch:10000|pcapiserv02.cern.ch:10000"
echo ===========================
echo $PATH
echo $ROOTSYS
echo $LD_LIBRARY_PATH
echo ==========================

root -b -x runProcess.C;

aliensh:[alice] [26] /alice/cern.ch/user/p/pchrist/ >
```

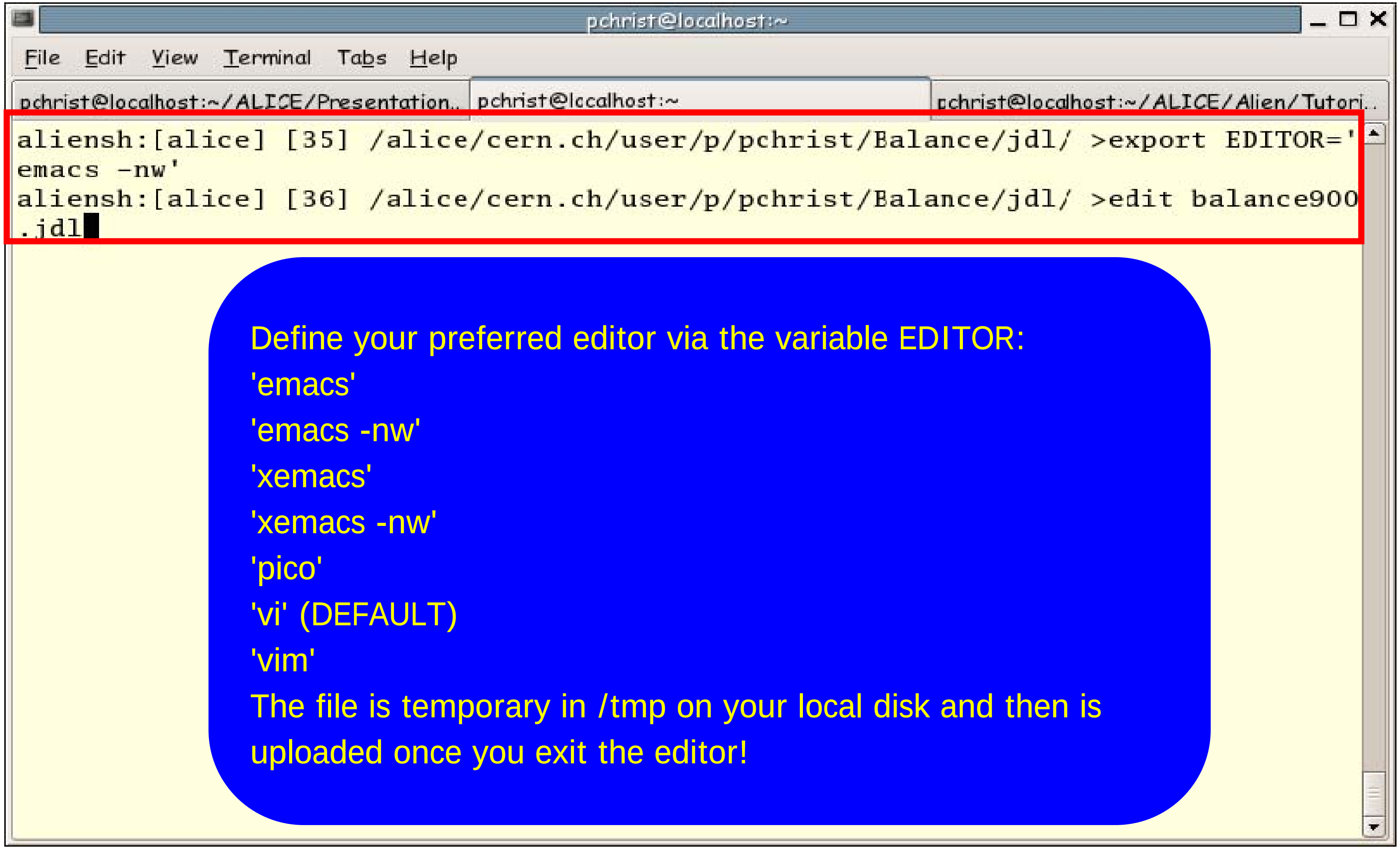
Shell (1) – Viewing the files II



pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
aliensh:[alice] [27] /alice/cern.ch/user/p/pchrist/ >more bin/batch.sh
#!/bin/bash
export GCLIENT_SERVER_LIST="pcapiserv01.cern.ch:10000|pcapiserv02.cern.ch:10000"
echo ===========================
echo $PATH
echo $ROOTSYS
echo $LD_LIBRARY_PATH
echo ==========================

root -b -x runProcess.C;

aliensh:[alice] [28] /alice/cern.ch/user/p/pchrist/ >
```

Shell (1) – Editing files


pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation..
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
aliensh:[alice] [35] /alice/cern.ch/user/p/pchrist/Balance/jdl/ >export EDITOR='
emacs -nw'
aliensh:[alice] [36] /alice/cern.ch/user/p/pchrist/Balance/jdl/ >edit balance900
.jdl
```

Define your preferred editor via the variable EDITOR:

'emacs'

'emacs -nw'

'xemacs'

'xemacs -nw'

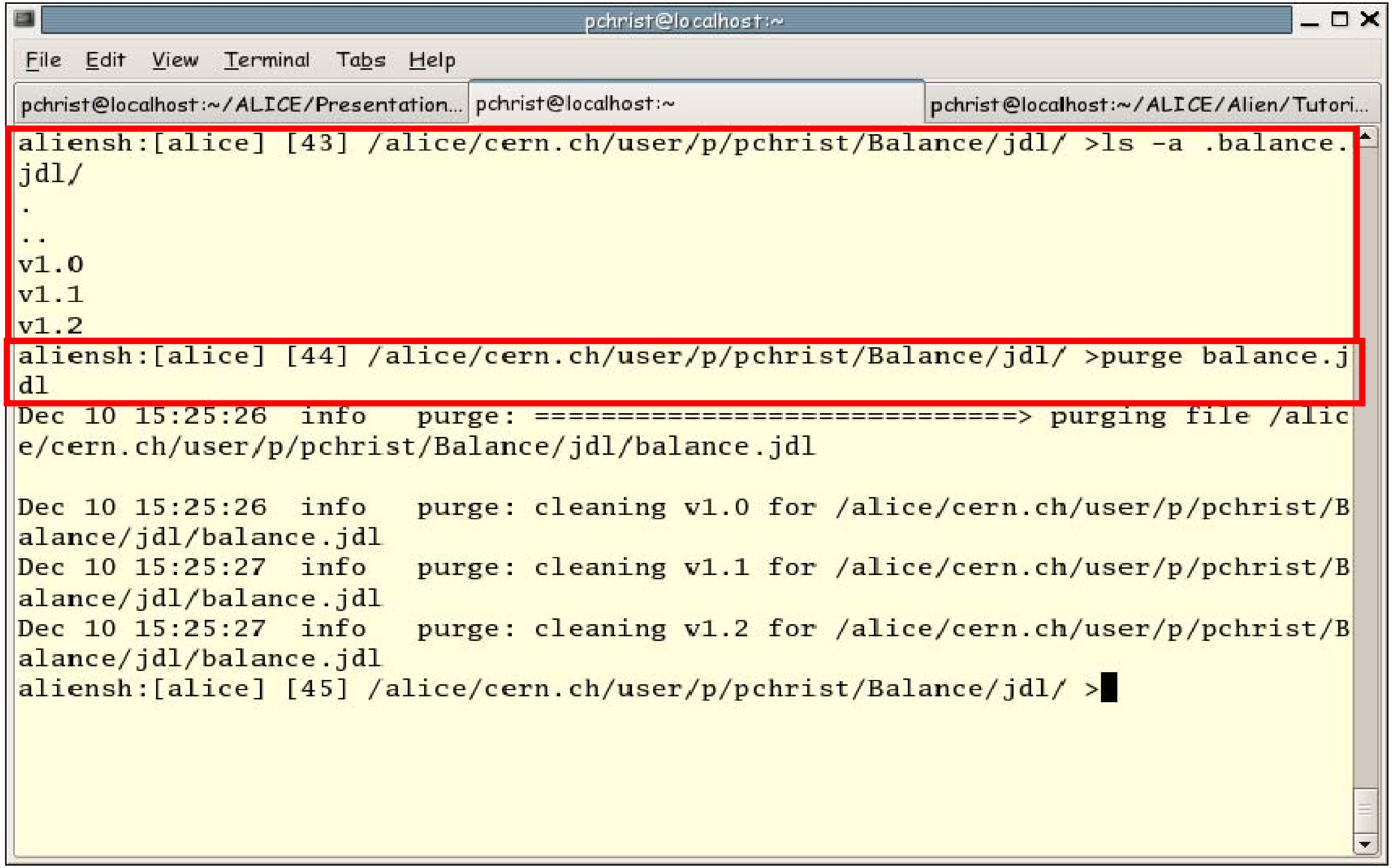
'pico'

'vi' (DEFAULT)

'vim'

The file is temporary in /tmp on your local disk and then is uploaded once you exit the editor!

Shell (1) – Clear old versions



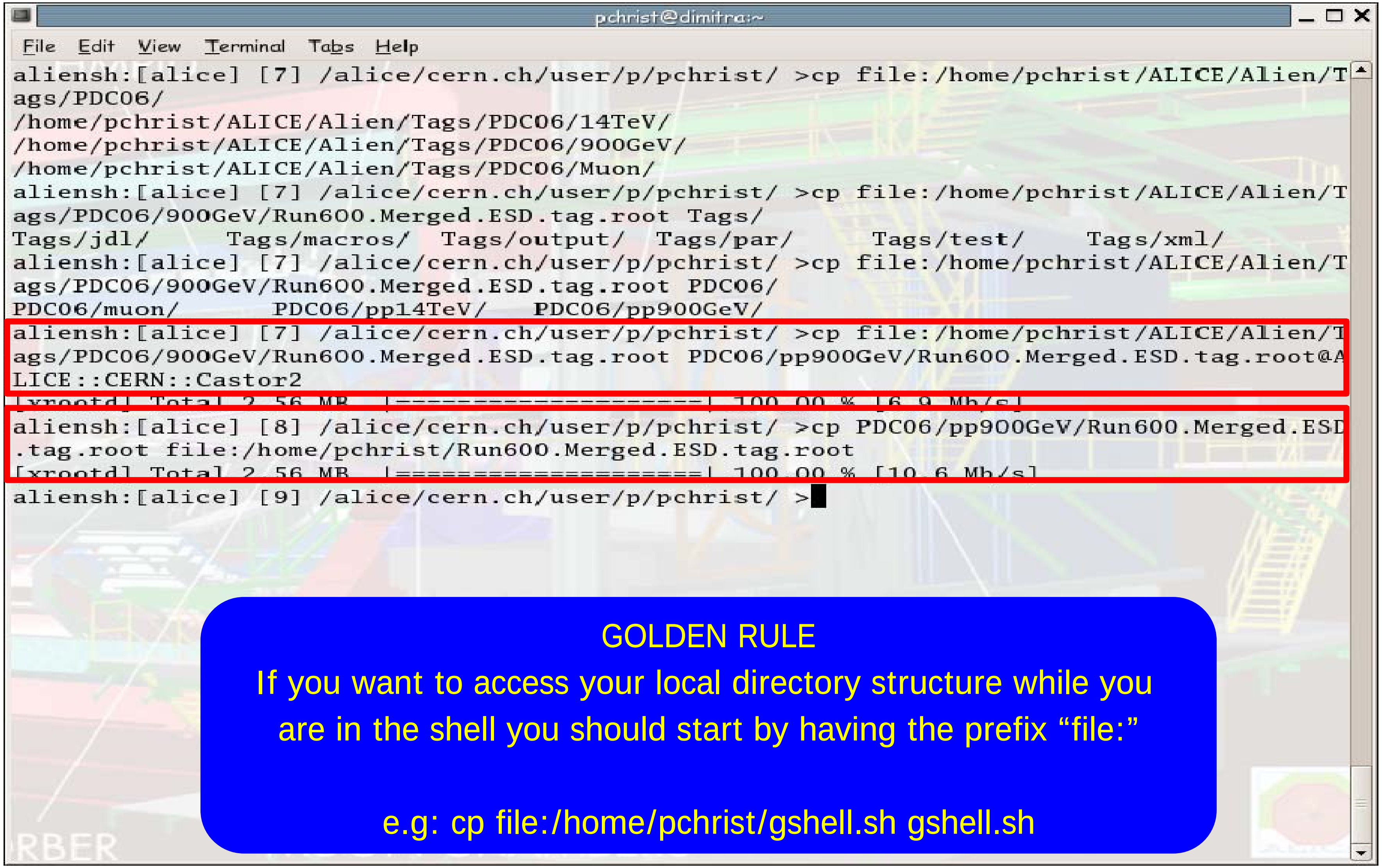
pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
aliensh:[alice] [43] /alice/cern.ch/user/p/pchrist/Balance/jdl/ >ls -a .balance.
jdl/
.
..
v1.0
v1.1
v1.2
aliensh:[alice] [44] /alice/cern.ch/user/p/pchrist/Balance/jdl/ >purge balance.j
dl
Dec 10 15:25:26  info   purge: ===============================> purging file /alic
e/cern.ch/user/p/pchrist/Balance/jdl/balance.jdl

Dec 10 15:25:26  info   purge: cleaning v1.0 for /alice/cern.ch/user/p/pchrist/B
alance/jdl/balance.jdl
Dec 10 15:25:27  info   purge: cleaning v1.1 for /alice/cern.ch/user/p/pchrist/B
alance/jdl/balance.jdl
Dec 10 15:25:27  info   purge: cleaning v1.2 for /alice/cern.ch/user/p/pchrist/B
alance/jdl/balance.jdl
aliensh:[alice] [45] /alice/cern.ch/user/p/pchrist/Balance/jdl/ >
```

Shell (1) – Try it out

- Check your user name by typing whoami.
- List the contents of your home directory.
- Check the working directory.
- Create the following directory structure:
  - $HOME/bin
  - $HOME/Tutorial/XML/jdl
  - $HOME/Tutorial/XML/par
  - $HOME/Tutorial/XML/output
  - $HOME/Tutorial/XML/selectors
  - $HOME/Tutorial/XML/macros
- Get the information of the file: /alice/cern.ch/user/p/pchrist/Tutorial/LOCAL/AliAnalysisTaskPt.cxx

Shell (2) – Copying files from/to the F.C.



```
aliensh:[alice] [7] /alice/cern.ch/user/p/pchrist/ >cp file:/home/pchrist/ALICE/Alien/T
ags/PDC06/
/home/pchrist/ALICE/Alien/Tags/PDC06/14TeV/
/home/pchrist/ALICE/Alien/Tags/PDC06/900GeV/
/home/pchrist/ALICE/Alien/Tags/PDC06/Muon/
aliensh:[alice] [7] /alice/cern.ch/user/p/pchrist/ >cp file:/home/pchrist/ALICE/Alien/T
ags/PDC06/900GeV/Run600.Merged.ESD.tag.root Tags/
Tags/jdl/      Tags/macros/  Tags/output/  Tags/par/     Tags/test/     Tags/xml/
aliensh:[alice] [7] /alice/cern.ch/user/p/pchrist/ >cp file:/home/pchrist/ALICE/Alien/T
ags/PDC06/900GeV/Run600.Merged.ESD.tag.root PDC06/
PDC06/muon/        PDC06/pp14TeV/    PDC06/pp900GeV/
aliensh:[alice] [7] /alice/cern.ch/user/p/pchrist/ >cp file:/home/pchrist/ALICE/Alien/T
ags/PDC06/900GeV/Run600.Merged.ESD.tag.root PDC06/pp900GeV/Run600.Merged.ESD.tag.root@A
LICE::CERN::Castor2
[xrootd] Total 2.56 MB  |====================| 100.00 % [6.9 Mb/s]
aliensh:[alice] [8] /alice/cern.ch/user/p/pchrist/ >cp PDC06/pp900GeV/Run600.Merged.ESD
.tag.root file:/home/pchrist/Run600.Merged.ESD.tag.root
[xrootd] Total 2.56 MB  |====================| 100.00 % [10.6 Mb/s]
aliensh:[alice] [9] /alice/cern.ch/user/p/pchrist/ >
```

GOLDEN RULE

If you want to access your local directory structure while you are in the shell you should start by having the prefix “file:”

e.g: cp file:/home/pchrist/gshell.sh gshell.sh

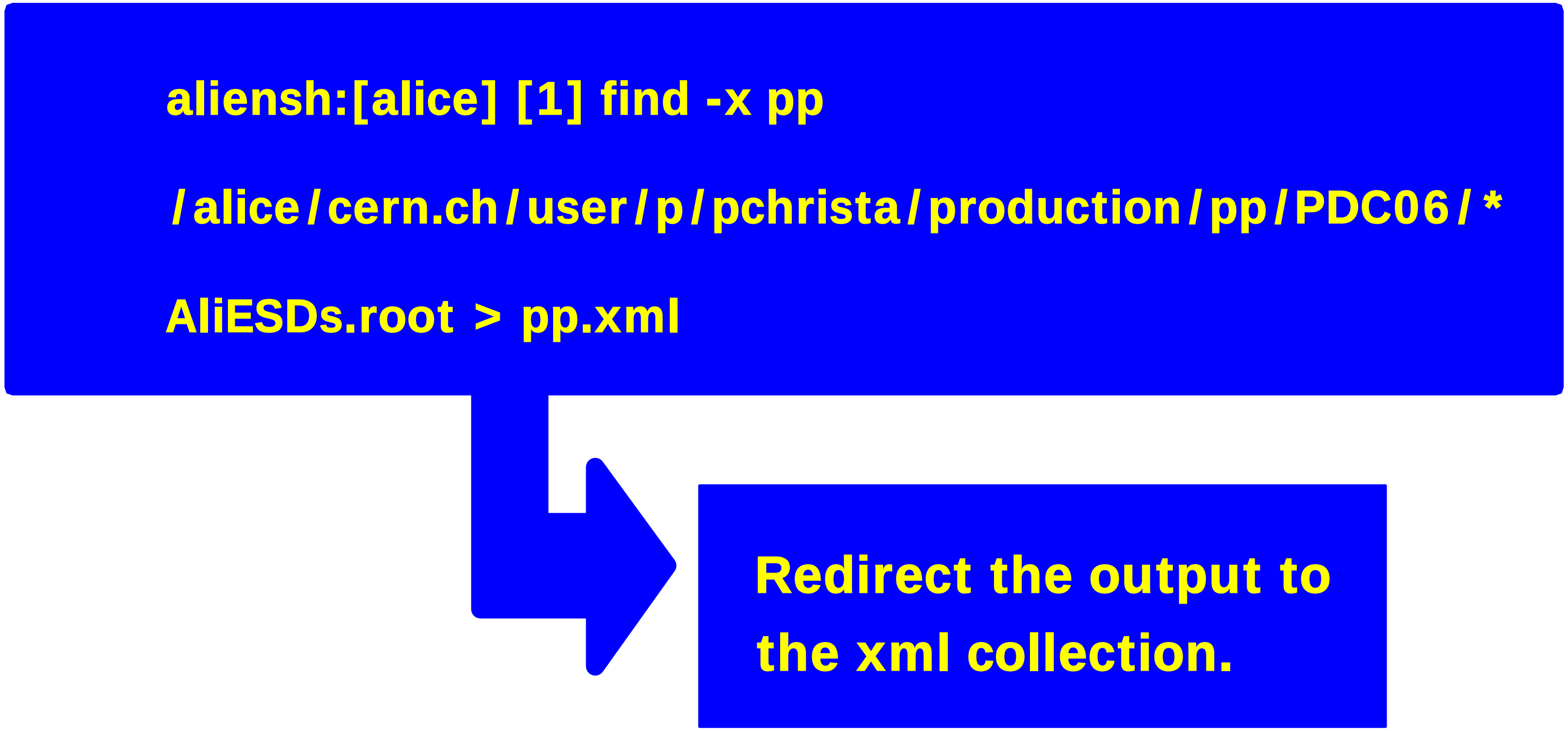
Shell (2) – File catalogue structure

- The path name will be:
  - for ‘real’ data: /data/<Year>/<AcceleratorPeriod>/<RunNumber>/
  - for simulated data: /sim/<Year>/<ProductionType>/<RunNumber>/
- Subdirectories will be called:
  - Raw/
  - cond/
  - reco/<PassX>/ESD/
  - reco/<PassX>/AOD/
  - ...

MARKUS OLDENBURG
AN INTERNAL NOTE
IS ON THE WAY

- File names will look like this: <xxxx>.AliESD.root
- For further information see:
  - http://indico.cern.ch/conferenceDisplay.py?confId=3280
  - http://cern.ch/Oldenburg/MetaData/MetaData.doc

Shell (2) – Querying the F.C. I



```
aliensh:[alice] [1] find -x pp
/alice/cern.ch/user/p/pchrista/production/pp/PDC06/*
AliESDs.root > pp.xml
```

Redirect the output to the xml collection.

Shell (2) – Querying the F.C. II

```
aliensh:[alice] [1] find -x pp

/alice/data/2008/LHC08a/*/reco/Pass3/*

AliESDs.root

Run:collision_system=”pp” and

Run:stop<"2008-03-20 10:20:33" and

Run:start>"2008-03-19" > pp.xml
```

Shell (2) – Try it out I

- Create the following directory structure locally:
  - $HOME/AliEn/PDC06/001 and $HOME/AliEn/PDC06/002
  - $HOME/AliEn/Tags
  - $HOME/AliEn/Local
  - $HOME/AliEn/Interactive
  - $HOME/AliEn/Batch
- Copy the following files to your local $HOME/AliEn/Local:
  - /alice/cern.ch/user/p/pchrist/Tutorial/LOCAL/ESD.par
  - /alice/cern.ch/user/p/pchrist/Tutorial/LOCAL/ANALYSIS_NEW.par
  - /alice/cern.ch/user/p/pchrist/Tutorial/LOCAL/AliAnalysisTaskPt.h
  - /alice/cern.ch/user/p/pchrist/Tutorial/LOCAL/AliAnalysisTaskPt.cxx
  - /alice/cern.ch/user/p/pchrist/Tutorial/LOCAL/runProcess.C
  - /alice/cern.ch/user/p/pchrist/Tutorial/LOCAL/demoLocal.C

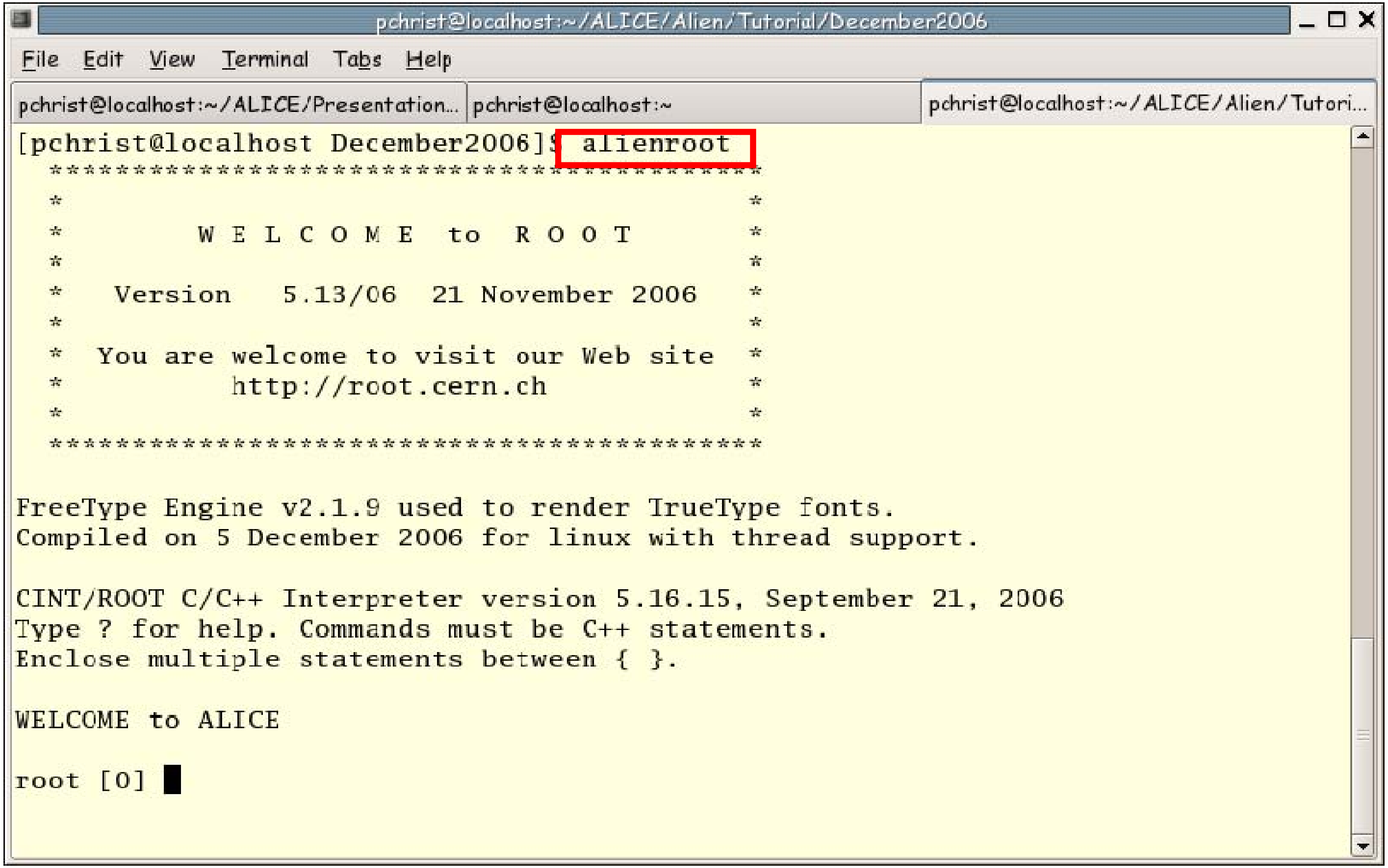
Shell (2) – Try it out II

- Copy the following files to your local $HOME/AliEn/Tags:
  - /alice/cern.ch/user/p/pchrist/Tutorial/TAGS/ESD.par
  - /alice/cern.ch/user/p/pchrist/Tutorial/TAGS/CreateTags.C
  - /alice/cern.ch/user/p/pchrist/Tutorial/TAGS/runProcess.C
- Copy the following files to $HOME/AliEn/PDC06/001 and $HOME/AliEn/PDC06/002:
  - /alice/cern.ch/user/p/pchrist/Tutorial/PDC06/001/AliESDs.root
  - /alice/cern.ch/user/p/pchrist/Tutorial/PDC06/002/AliESDs.root
- Query the f.c. and get all the tag files (*.tag.root) under: /alice/cern.ch/user/p/pchrist/Tutorial/PDC06/*
  - Get the output on your terminal.
  - Redirect the results to the tag10.xml collection.
- Repeat the previous exercise limiting the number of output files to 5 (find -l 5 ...) and copy the xml to your local $HOME/AliEn/Interactive.

Production status

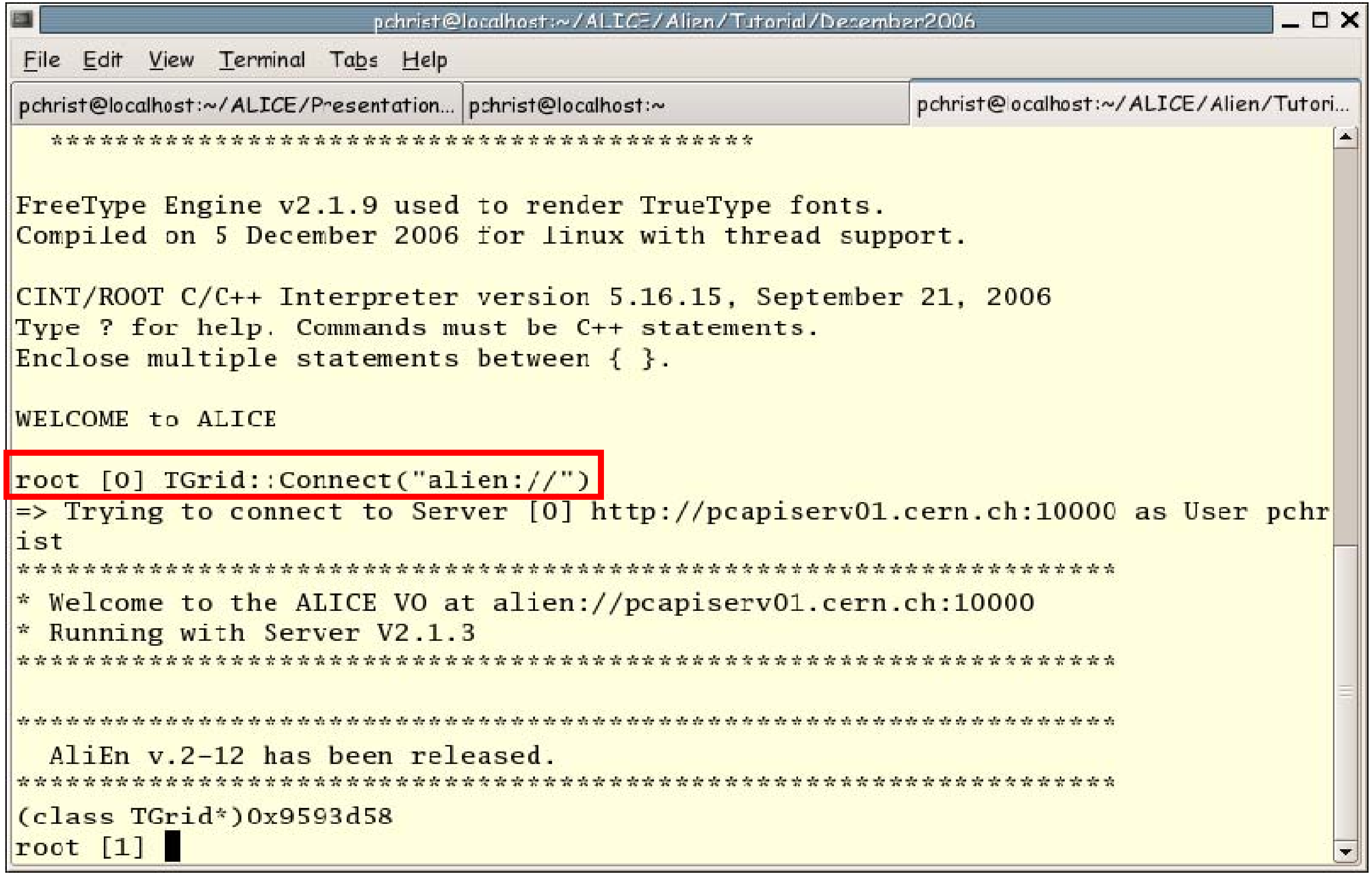
- p+p min bias @ 14TeV --- N_{Events} ~ 20M:
  - /alice/cern.ch/user/a/aliprod/prod2006_2/output_pp/ (5.4M)
  - /alice/sim/2006/pp_minbias/ (~12M)
  - /alice/sim/2006/pp_x_vertex_1cm/ (1.1M)
  - /alice/sim/2006/pp_x_vertex_05cm/ (1.1M)
  - /alice/sim/2006/pp_minbias_full/ (1.1M)
    - All RUNS have been tested and merged tag files have been produced at the RUN level for all RunIds.
- p+p min bias @ 900GeV --- N_{Events} = 200K:
  - /alice/sim/2006/pp_900GeV/
    - All RUNS have been tested and merged tag files have been produced at the RUN level for all RunIds.
- Muon events --- N_{Events} = 864K:
  - /alice/sim/2006/muon/ (64K)
  - /alice/sim/2006/muon_signle/ (800K)

ROOT – ROOT API


pchrist@localhost:~/ALICE/Alien/Tutorial/December2006
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
[pchrist@localhost December2006]$ alienroot
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.13/06  21 November 2006   *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

FreeType Engine v2.1.9 used to render TrueType fonts.
Compiled on 5 December 2006 for linux with thread support.

CINT/ROOT C/C++ Interpreter version 5.16.15, September 21, 2006
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.

WELCOME to ALICE

root [0] 
```

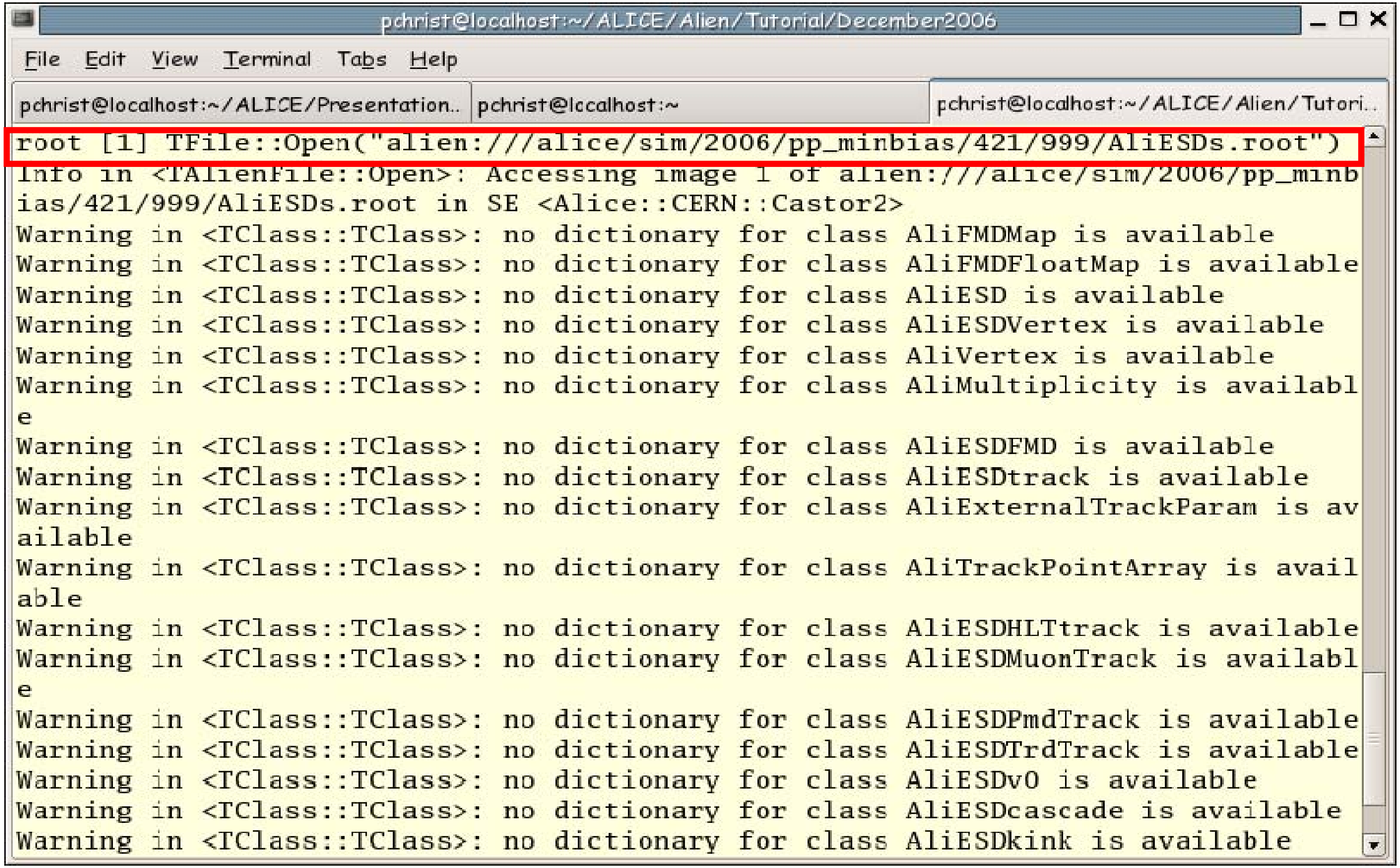
ROOT - Connecting



pchrist@localhost:~/ALICE/Alien/Tutorial/December2006
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
  *******************************************

FreeType Engine v2.1.9 used to render TrueType fonts.
Compiled on 5 December 2006 for linux with thread support.

CINT/ROOT C/C++ Interpreter version 5.16.15, September 21, 2006
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.

WELCOME to ALICE

root [0] TGrid::Connect("alien://")
=> Trying to connect to Server [0] http://pcapiserv01.cern.ch:10000 as User pchr
ist
********************************************************************
* Welcome to the ALICE VO at alien://pcapiserv01.cern.ch:10000
* Running with Server V2.1.3
********************************************************************


********************************************************************
  AliEn v.2-12 has been released.
********************************************************************
(class TGrid*)0x9593d58
root [1] 
```

ROOT – Accessing a GRID file



pchrist@localhost:~/ALICE/Alien/Tutorial/December2006
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation..
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori..

```
root [1] TFile::Open("alien:///alice/sim/2006/pp_minbias/421/999/AliESDs.root")
Info in <TAlienFile::Open>: Accessing image 1 of alien:///alice/sim/2006/pp_minb
ias/421/999/AliESDs.root in SE <Alice::CERN::Castor2>
Warning in <TClass::TClass>: no dictionary for class AliFMDMap is available
Warning in <TClass::TClass>: no dictionary for class AliFMDFloatMap is available
Warning in <TClass::TClass>: no dictionary for class AliESD is available
Warning in <TClass::TClass>: no dictionary for class AliESDVertex is available
Warning in <TClass::TClass>: no dictionary for class AliVertex is available
Warning in <TClass::TClass>: no dictionary for class AliMultiplicity is availabl
e
Warning in <TClass::TClass>: no dictionary for class AliESDFMD is available
Warning in <TClass::TClass>: no dictionary for class AliESDtrack is available
Warning in <TClass::TClass>: no dictionary for class AliExternalTrackParam is av
ailable
Warning in <TClass::TClass>: no dictionary for class AliTrackPointArray is avail
able
Warning in <TClass::TClass>: no dictionary for class AliESDHLTtrack is available
Warning in <TClass::TClass>: no dictionary for class AliESDMuonTrack is availabl
e
Warning in <TClass::TClass>: no dictionary for class AliESDPmdTrack is available
Warning in <TClass::TClass>: no dictionary for class AliESDTrdTrack is available
Warning in <TClass::TClass>: no dictionary for class AliESDv0 is available
Warning in <TClass::TClass>: no dictionary for class AliESDcascade is available
Warning in <TClass::TClass>: no dictionary for class AliESDkink is available
```

ROOT – Problems with gcc versions

- AliEn software comes with a precompiled library with gcc 3.2.3 which is also shipped with AliEn.
- If you are using a different version of gcc (check it with gcc –v) then do the following:
  - cd $ALIEN/api/src
  - ./recompile.gapi
  - cd $ALIEN/api/lib
  - Copy all the libgapiUI.so.2* files to the libgapiUI.so.3*
- Or link your gcc to the version shipped with AliEn and compile everything with this.

ROOT – Try it out I

- Go to your $ROOTSYS and change the cfg.sh file – change the location where you installed the grid software (it should be e.g. /home/trn2301/alice/AliEn).
- Source the script and then type make and make map.
- Write the following script, name it alienroot, make it executable, place it in your $HOME/bin directory and add it in your $PATH.

```
#!/bin/bash
export ALIEN=/home/trn2301/alien
export ROOTSYS=/home/trn2301/root
export PATH=$ROOTSYS/bin:$PATH
export LD_LIBRARY_PATH=$ROOTSYS/lib:$LD_LIBRARY_PATH:$ALIEN/api/lib

if [ -e /tmp/gclient_env_$UID ]; then
     source /tmp/gclient_env_$UID;
     root.exe $*
fi
```

ROOT – Try it out II

- Once finished, change the file to executable, get a token, type alienroot and then:
  - root [0] TGrid::Connect(“alien://”);
- If it works without error messages you have installed everything successfully. If not then go back to the previous page of this tutorial.

Framework – New analysis framework

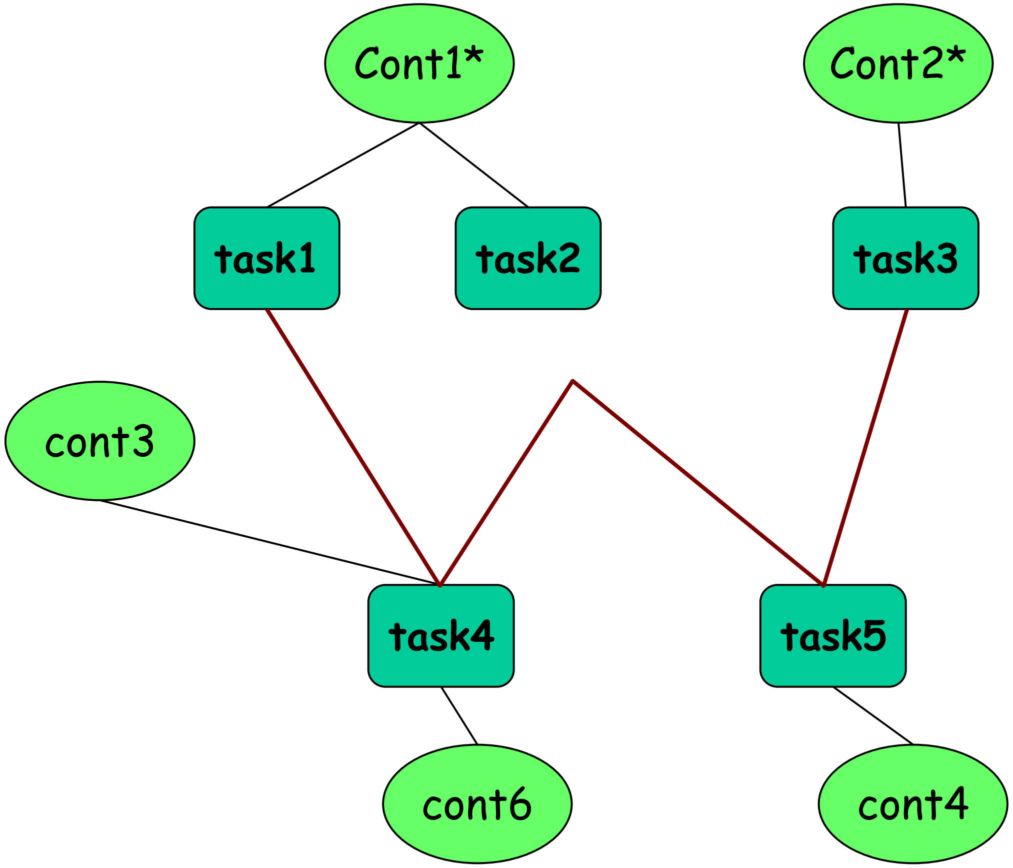
- AliAnalysisDataContainer:
  - Class that allows the user to define the basic input/output containers.
  - Three types of containers: input, transient and output.
- AliAnalysisTask:
  - Implementation of the actual analysis code that processes input data.
- AliAnalysisManager:
  - Definition of all data containers that will assembly the analysis.
  - Definition of tasks.
  - Definition of the relationships between the tasks and the containers.

Andrei Gheata

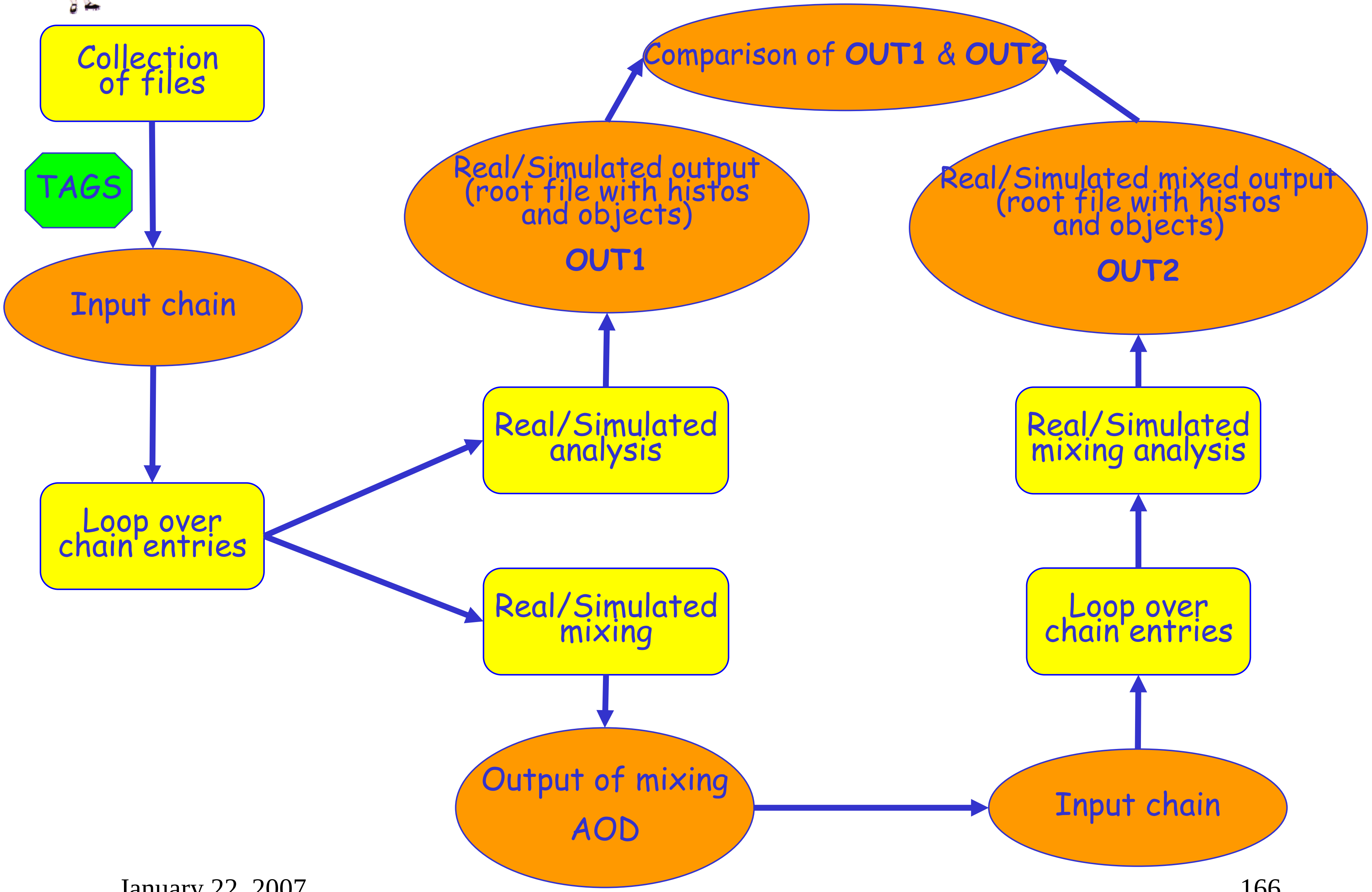
http://indico.cern.ch/materialDisplay.py?contribId=19&amp;sessionId=3&amp;materialId=slides&amp;confId=a056304

Framework – Data flow structure

AliAnalysisManager
TObjArray *fContainers
TObjArray *fTasks



Framework – A practical example



Collection of files

TAGS

Input chain

Loop over chain entries

Real/Simulated analysis

Real/Simulated mixing

Output of mixing

AOD

Real/Simulated output (root file with histos and objects)

OUT1

Comparison of **OUT1** & **OUT2**

Input chain

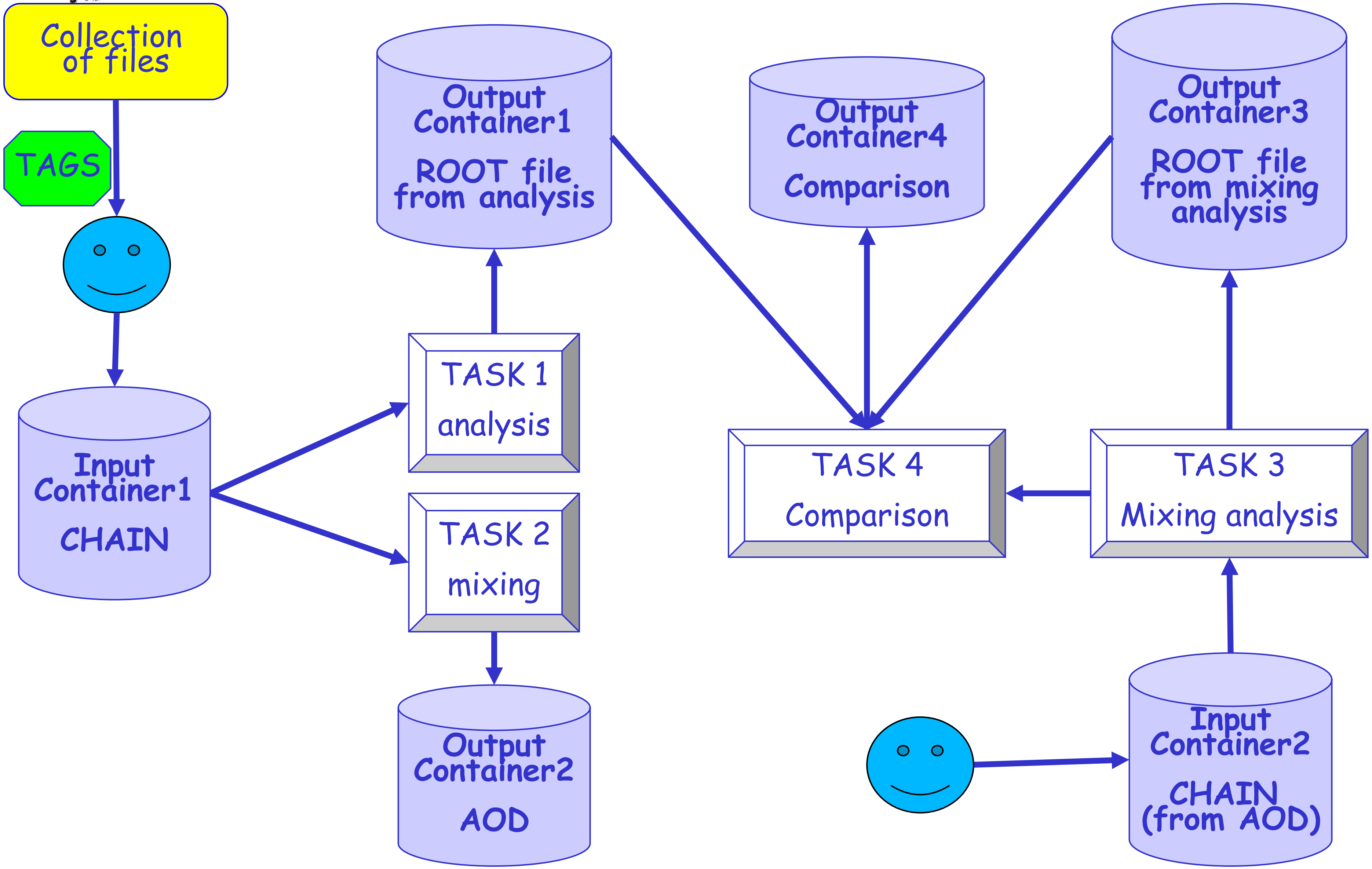
Loop over chain entries

Real/Simulated mixing analysis

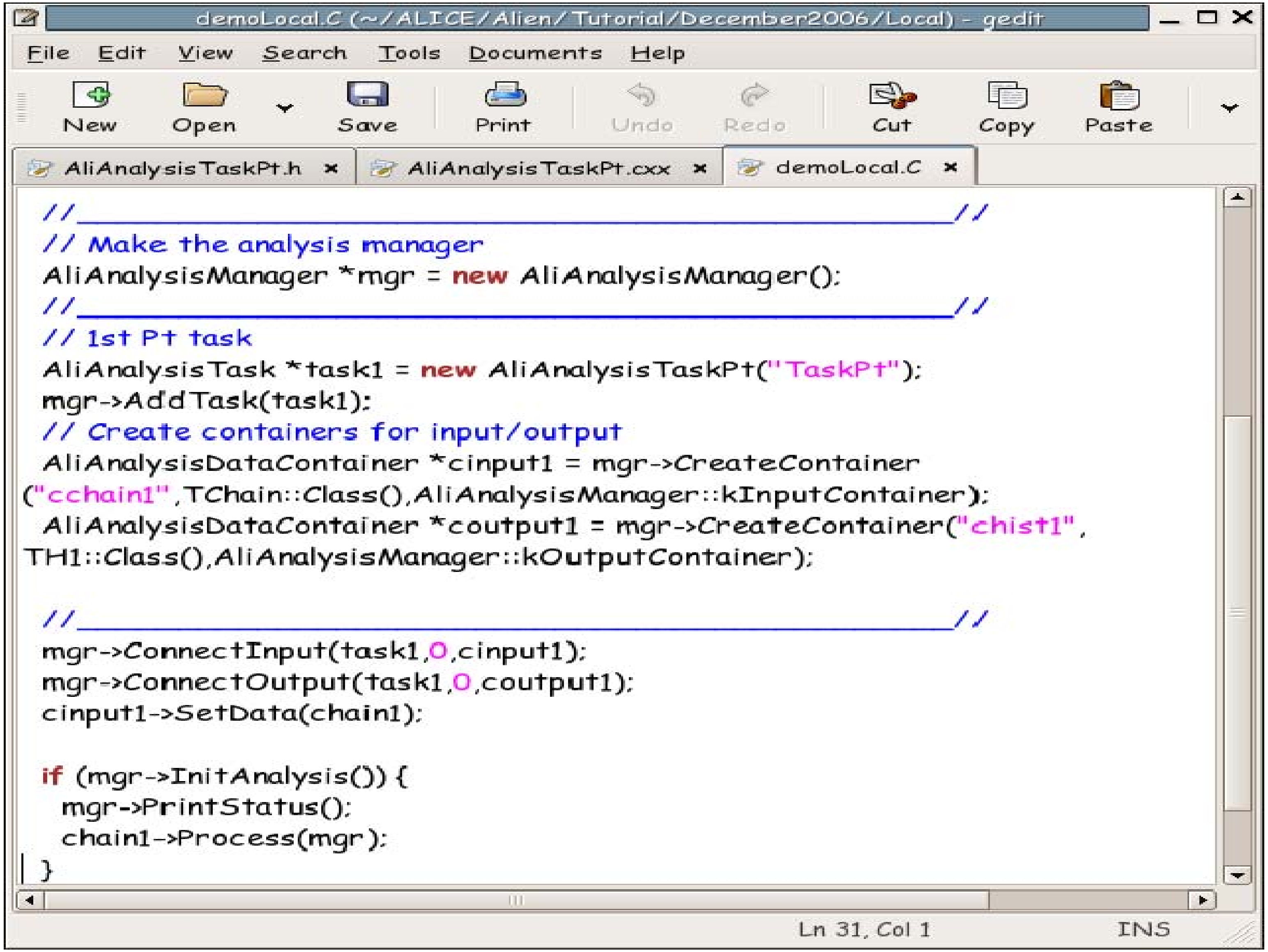
Real/Simulated mixed output (root file with histos and objects)

OUT2

Framework – Integration in the framework

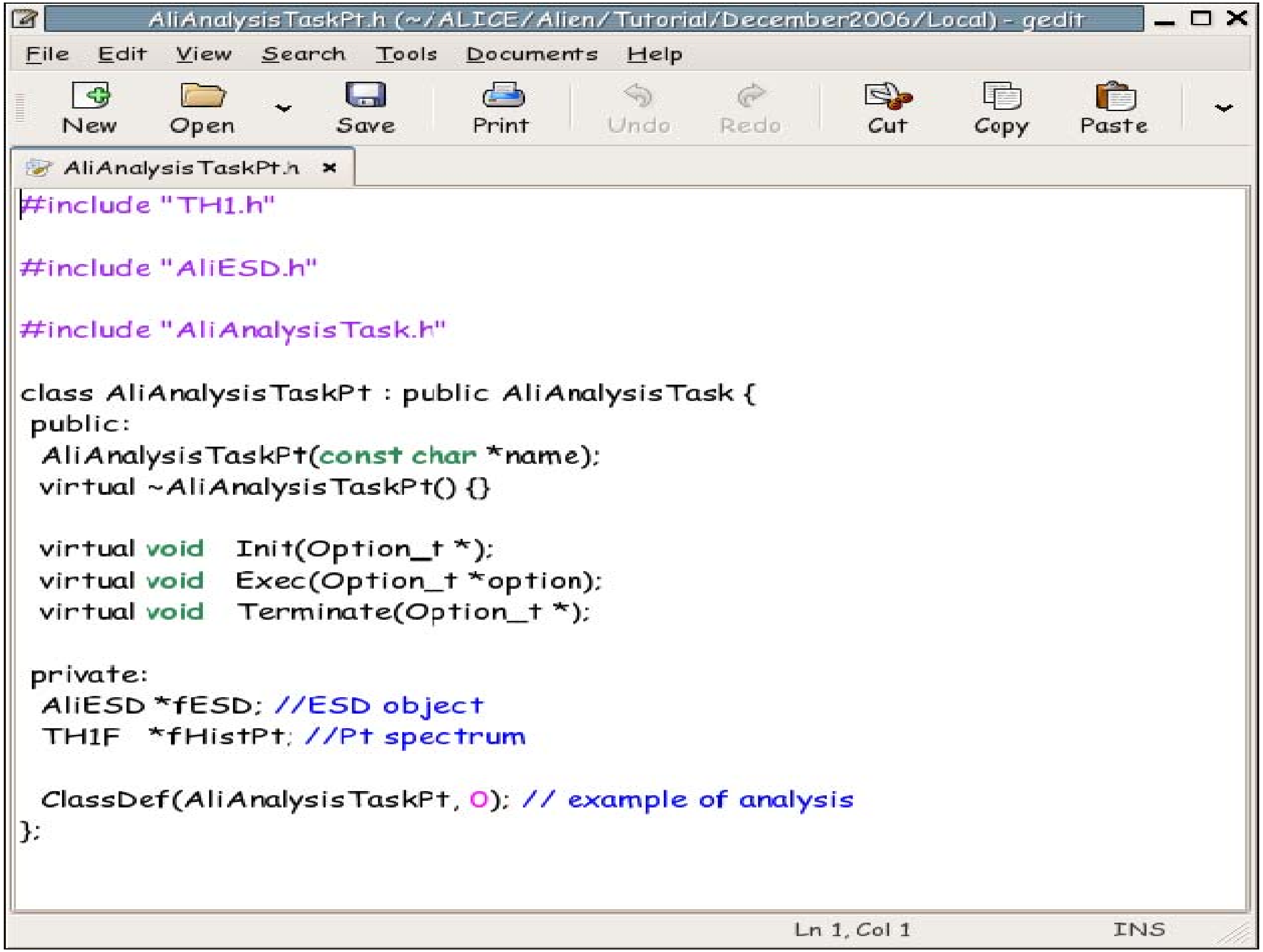


Framework – Example of a manager


demoLocal.C (~/ALICE/Alien/Tutorial/December2006/Local) - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste
AliAnalysisTaskPt.h
AliAnalysisTaskPt.cxx
demoLocal.C
Ln 31, Col 1
INS

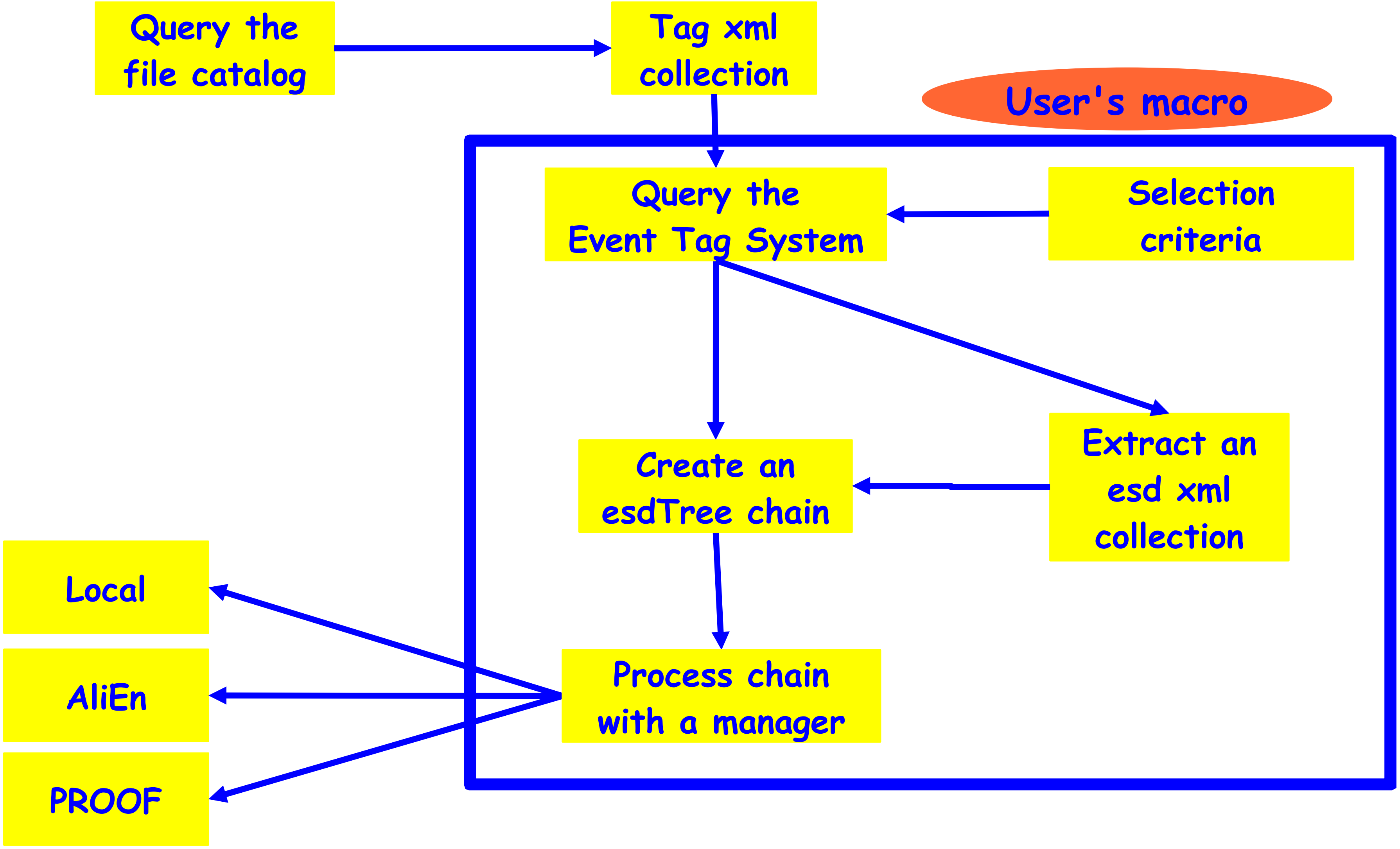
```
 //________________________________________________________________//
 // Make the analysis manager
 AliAnalysisManager *mgr = new AliAnalysisManager();
 //________________________________________________________________//
 // 1st Pt task
 AliAnalysisTask *task1 = new AliAnalysisTaskPt("TaskPt");
 mgr->AddTask(task1);
 // Create containers for input/output
 AliAnalysisDataContainer *cinput1 = mgr->CreateContainer
("cchain1",TChain::Class(),AliAnalysisManager::kInputContainer);
 AliAnalysisDataContainer *coutput1 = mgr->CreateContainer("chist1",
TH1::Class(),AliAnalysisManager::kOutputContainer);

 //________________________________________________________________//
 mgr->ConnectInput(task1,0,cinput1);
 mgr->ConnectOutput(task1,0,coutput1);
 cinput1->SetData(chain1);

 if (mgr->InitAnalysis()) {
   mgr->PrintStatus();
   chain1->Process(mgr);
 }
```

Framework – Example of a task

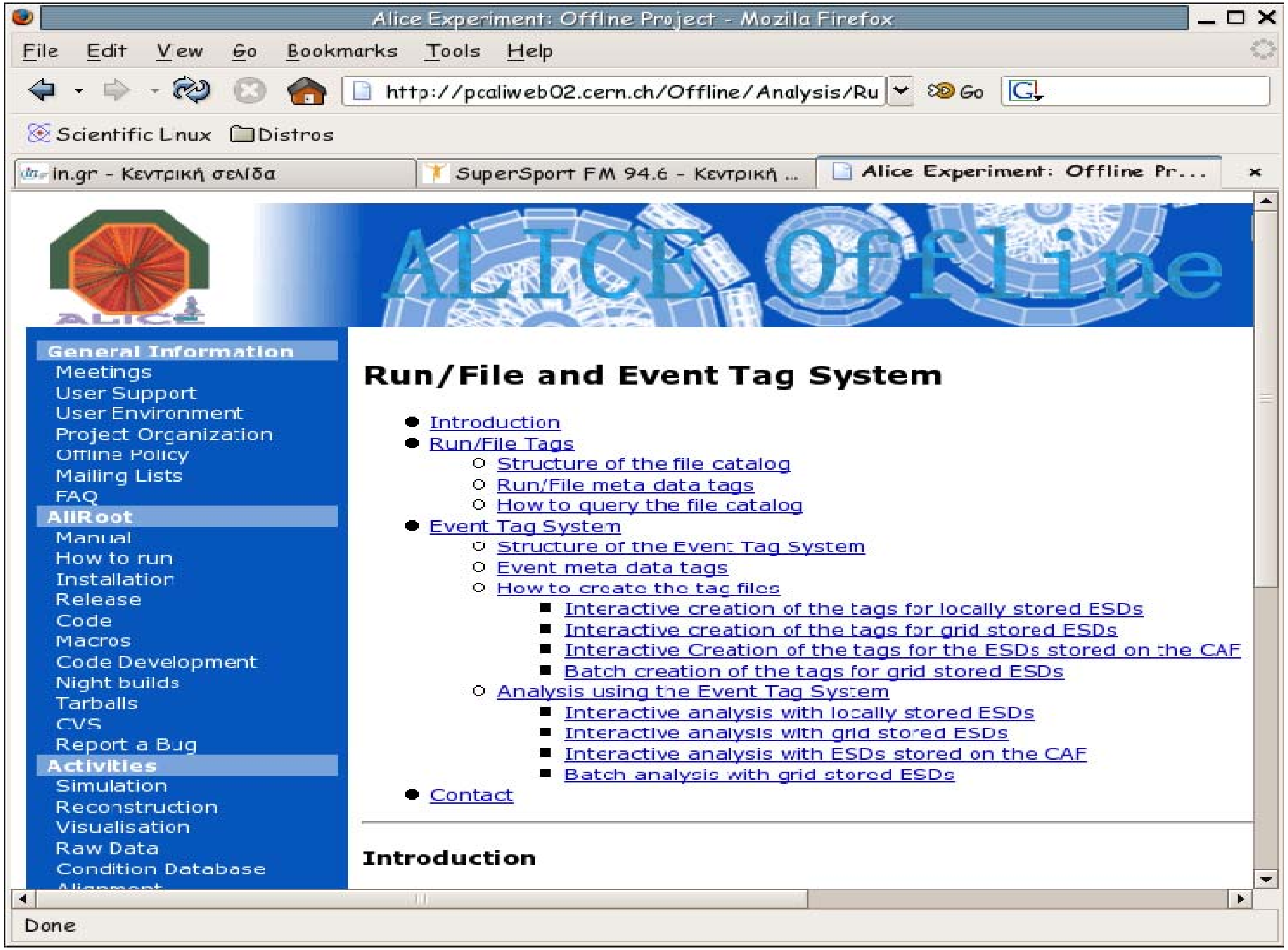

AliAnalysisTaskPt.h (~/ALICE/Alien/Tutorial/December2006/Local) - gedit
File Edit View Search Tools Documents Help
New
Open
Save
Print
Undo
Redo
Cut
Copy
Paste
AliAnalysisTaskPt.h
Ln 1, Col 1
INS

```
#include "TH1.h"

#include "AliESD.h"

#include "AliAnalysisTask.h"

class AliAnalysisTaskPt : public AliAnalysisTask {
 public:
  AliAnalysisTaskPt(const char *name);
  virtual ~AliAnalysisTaskPt() {}

  virtual void   Init(Option_t *);
  virtual void   Exec(Option_t *option);
  virtual void   Terminate(Option_t *);

 private:
  AliESD *fESD; //ESD object
  TH1F   *fHistPt; //Pt spectrum

  ClassDef(AliAnalysisTaskPt, 0); // example of analysis
};
```

Analysis – Flow of the analysis procedure



Analysis – Event Tag System



Run/File and Event Tag System

- Introduction
- Run/File Tags
  - Structure of the file catalog
  - Run/File meta data tags
  - How to query the file catalog
- Event Tag System
  - Structure of the Event Tag System
  - Event meta data tags
  - How to create the tag files
    - Interactive creation of the tags for locally stored ESDs
    - Interactive creation of the tags for grid stored ESDs
    - Interactive Creation of the tags for the ESDs stored on the CAF
    - Batch creation of the tags for grid stored ESDs
  - Analysis using the Event Tag System
    - Interactive analysis with locally stored ESDs
    - Interactive analysis with grid stored ESDs
    - Interactive analysis with ESDs stored on the CAF
    - Batch analysis with grid stored ESDs
- Contact

Introduction

Local analysis – Creation of tag files

Setup par archive
Load the needed libraries

```
AliTagCreator *t = new AliTagCreator();
```

```
t->SetStorage(0);
```

```
t->ReadLocalCollection("/home/pchrist/PDC06/pp14TeV/");
```

```
t->MergeTags();
```

Local analysis – Local analysis with tags I

Setup par archive
Load the needed libraries

```
AliRunTagCuts *RunCuts = new AliRunTagCuts();
AliEventTagCuts *EvCuts = new AliEventTagCuts();
EvCuts->SetMultiplicityRange(0,1500);
```

```
AliTagAnalysis *TagAna = new AliTagAnalysis();
TagAna->ChainLocalTags(“.”);
```

```
analysischain = TagAna->QueryTags(RunCuts,EvCuts);
```

```
const char *selectorfile = "esdPt.C";
analysischain->Process(selectorfile);
```

The query of the Event Tag System can be done by:

❑Using the AliRunTagCuts and AliEventTagCuts objects

❑Using string statements (“(fEventTag.fNumberOfTracks > 0)&&(fEventTag.fNumberOfTracks < 1500)”)

Local analysis – Local analysis with tags II

Setup par archive
Load the needed libraries

```
AliRunTagCuts *RunCuts = new AliRunTagCuts();
AliEventTagCuts *EvCuts = new AliEventTagCuts();
EvCuts->SetMultiplicityRange(0,1500);
```

```
AliTagAnalysis *TagAna = new AliTagAnalysis();
TagAna->ChainLocalTags(“.”);
```

```
TChain *analysischain = TagAna->QueryTags(RunCuts,EvCuts);
```

```
AliAnalysisManager *manager = new AliAnalysisManager();
AliAnalysisTask *task = new AliAnalysisTaskPt(“TaskPt”);
manager->AddTask(task);
```

```
AliAnalysisDataContainer *cinput1 = manager-
>CreateContainer("cchain1",TChain::Class(),AliAnalysisManager::kInputContainer);
AliAnalysisDataContainer *coutput1 = manager->CreateContainer("chist1",
TH1::Class(),AliAnalysisManager::kOutputContainer);
```

```
manager->ConnectInput(task,0,cinput1);
manager->ConnectOutput(task,0,coutput1);
cinput1->SetData(chain1);
```

```
analysischain->Procees(manager);
```

Local analysis – Try it out

- Open your local $HOME/AliEn/Tags/CreateTags.C file and modify it accordingly:
  - Change the line where you define where you have the locally stored ESDs.
- Run it to create the tag files with alienroot.
- Delete the single tag files and stay with just the merged one.
- Go to your local $HOME/AliEn/Local directory and open the demoLocal.C file.
- Change the line where you define the location of the tag files and run the macro with alienroot.
- Impose some selection criteria and rerun the example.

Interactive analysis – Using the tags

Setup par archive
Load the needed libraries

```
TGrid::Connect("alien://”);
```

```
TAlienCollection* coll = TalienCollection::Open("tag100.xml");
TGridResult* TagResult = coll->GetGridResult("");
```

```
AliTagAnalysis *TagAna = new AliTagAnalysis();
TagAna->ChainGridTags(TagResult);
```

```
AliRauTagCuts *RunCuts = new AliRunTagCuts();
AliEventTagCuts *EvCuts = new AliEventTagCuts();
EvCuts->SetMultiplicityRange(0,1500);
```

```
TChain *analysischain = TagAna->QueryTags(RunCuts,EvCuts);
```

```
AliAnalysisManager *manager = new AliAnalysisManager();
AliAnalysisTask *task = new AliAnalysisTaskPt(“TaskPt”);
manager->AddTask(task);
```

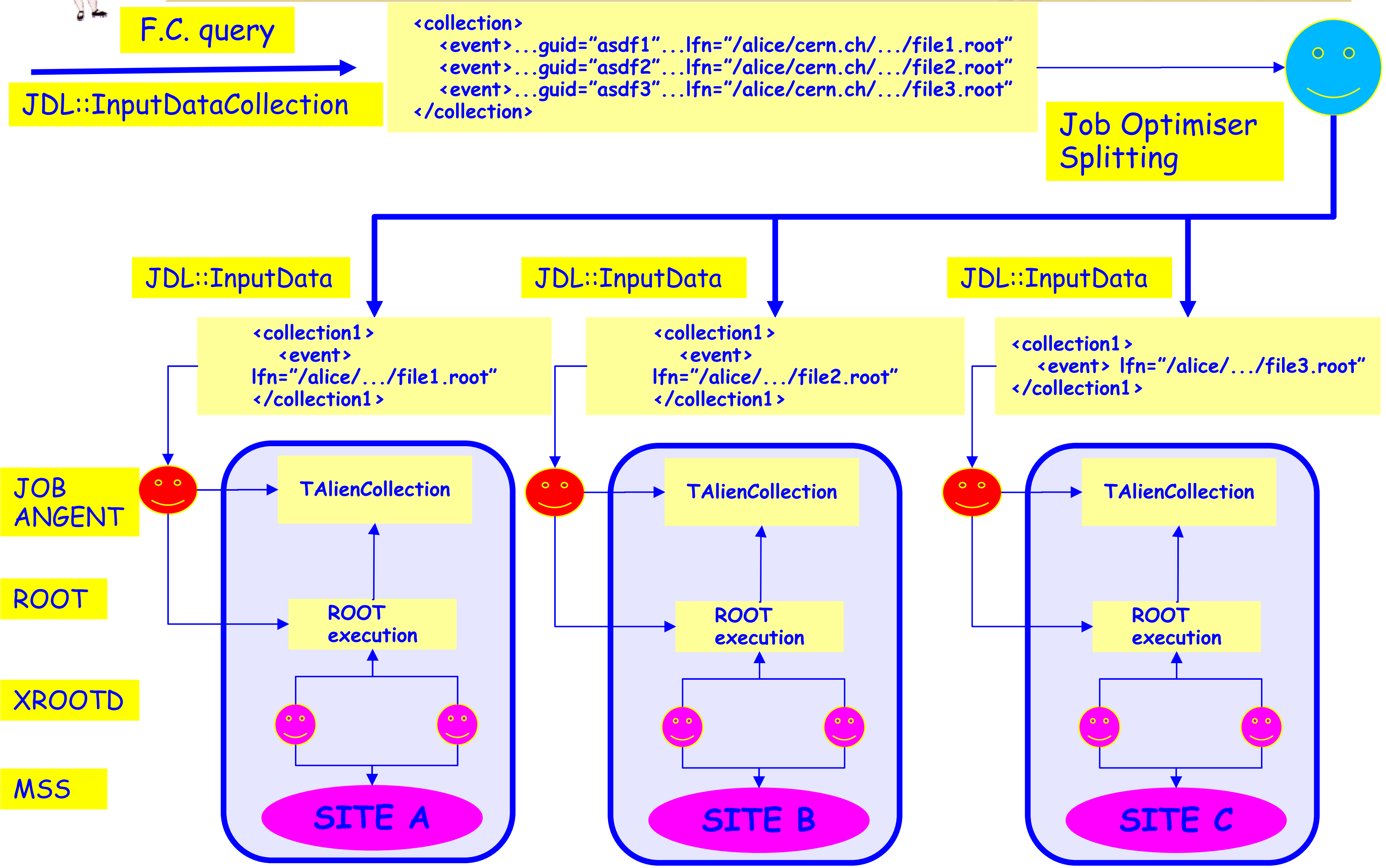
Same code as on two pages back

```
analysischain->Procees(manager);
```

Interactive analysis – Try it out

- Copy the following files to your local $HOME/AliEn/Interactive:
  - /alice/cern.ch/user/p/pchrist/Tutorial/INTERACTIVE/ESD.par
  - /alice/cern.ch/user/p/pchrist/Tutorial/INTERACTIVE/ANALYSIS_NEW.par
  - /alice/cern.ch/user/p/pchrist/Tutorial/INTERACTIVE/AliAnalysisTaskPt.h
  - /alice/cern.ch/user/p/pchrist/Tutorial/INTERACTIVE/AliAnalysisTaskPt.cxx
  - /alice/cern.ch/user/p/pchrist/Tutorial/INTERACTIVE/runProcess.C
  - /alice/cern.ch/user/p/pchrist/Tutorial/INTERACTIVE/demoInteractive.C
- Go to your local $HOME/AliEn/Interactive directory and open the demoInteractive.C file.
- Change the line where you define the tag collection and put the name of the file you created by querying the f.c (it should be tag10.xml).
- Run the macro with alienroot – Impose some selection criteria and rerun the example.

Batch analysis – Flow of batch analysis



F.C. query

JDL::InputDataCollection

```
<collection>
   <event>...guid="asdf1"...lfn="/alice/cern.ch/.../file1.root"
   <event>...guid="asdf2"...lfn="/alice/cern.ch/.../file2.root"
   <event>...guid="asdf3"...lfn="/alice/cern.ch/.../file3.root"
</collection>
```

Job Optimiser
Splitting

JDL::InputData

```
<collection1>
   <event>
lfn="/alice/.../file1.root"
</collection1>
```

JDL::InputData

```
<collection1>
   <event>
lfn="/alice/.../file2.root"
</collection1>
```

JDL::InputData

```
<collection1>
   <event> lfn="/alice/.../file3.root"
</collection1>
```

JOB
ANGENT

ROOT

XROOTD

MSS

TAlienCollection

ROOT
execution

SITE A

TAlienCollection

ROOT
execution

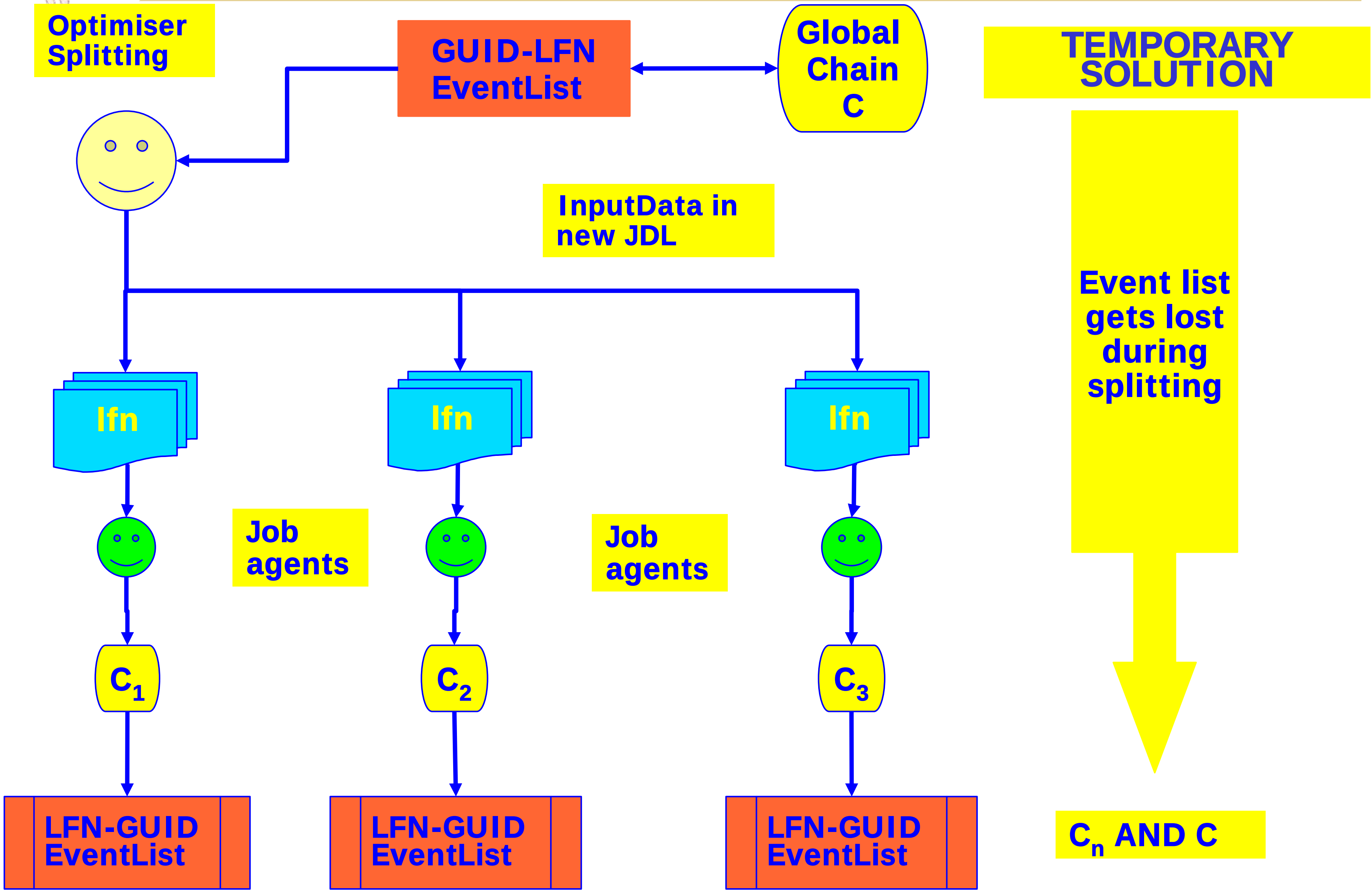
SITE B

TAlienCollection

ROOT
execution

SITE C

Batch analysis – Analysis flow with tags I



Optimiser Splitting

GUID-LFN EventList

Global Chain C

TEMPORARY SOLUTION

InputData in new JDL

lfn

lfn

lfn

Job agents

Job agents

C_1

C_2

C_3

Event list gets lost during splitting

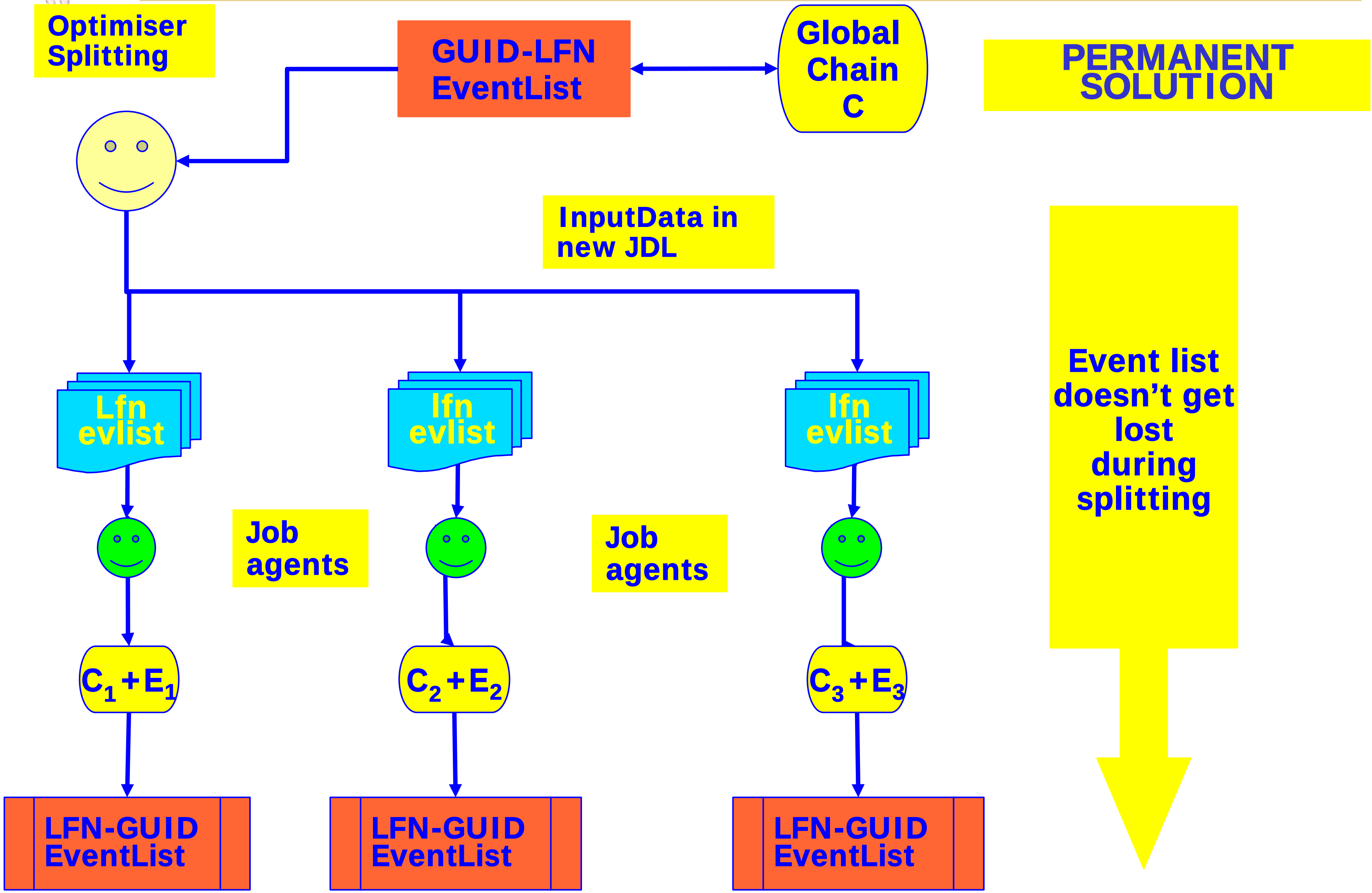
LFN-GUID EventList

LFN-GUID EventList

LFN-GUID EventList

C_n AND C

Batch analysis – Analysis flow with tags II



Optimiser Splitting

GUID-LFN EventList

Global Chain C

PERMANENT SOLUTION

InputData in new JDL

Lfn evlist

lfn evlist

lfn evlist

Job agents

Job agents

C_1+E_1

C_2+E_2

C_3+E_3

LFN-GUID EventList

LFN-GUID EventList

LFN-GUID EventList

Event list doesn’t get lost during splitting

Batch analysis – Creating a new xml file

Setup par archive
Load the needed libraries

You need to use the tags in a batch session because:

□ They reduce your analysis time which allows you to lower your TTL (see next slides) and thus make sure that your job starts early enough (jobs are ordered by TTL).

□ They provide the analyzed data in the proper format (TChain + TEntryLists) in a totally transparent way.

```
TagAna->CreateXMLCollection(“global”,RunCuts,EvCuts);
```

The old xml collection (tag.xml) has information about the tag files that are qoing to be queried.
The new xml collection (global.xml) has information about the ESDs that are going to be analyzed.

Batch analysis – Files for batch analysis

- Executable
- Par file
- Macro
- Selectors
- xml collection
- jdl

Batch analysis - Executable

```
#!/bin/bash

echo
===========================
echo $PATH
echo $ROOTSYS
echo $LD_LIBRARY_PATH
echo ==========================

root -b -x Analysis.C;
```

IT SHOULD BE STORED UNDER

$HOME/bin IN THE FILE CATALOG!!!

Batch analysis - Macro

- Setup the par file – compile and load the libESD.so (or any necessary library that is needed for the analysis).
- Get the xml collection.
- Convert the collection to a list of files.
- Process the chain with the selector or an AliAnalysisManager.

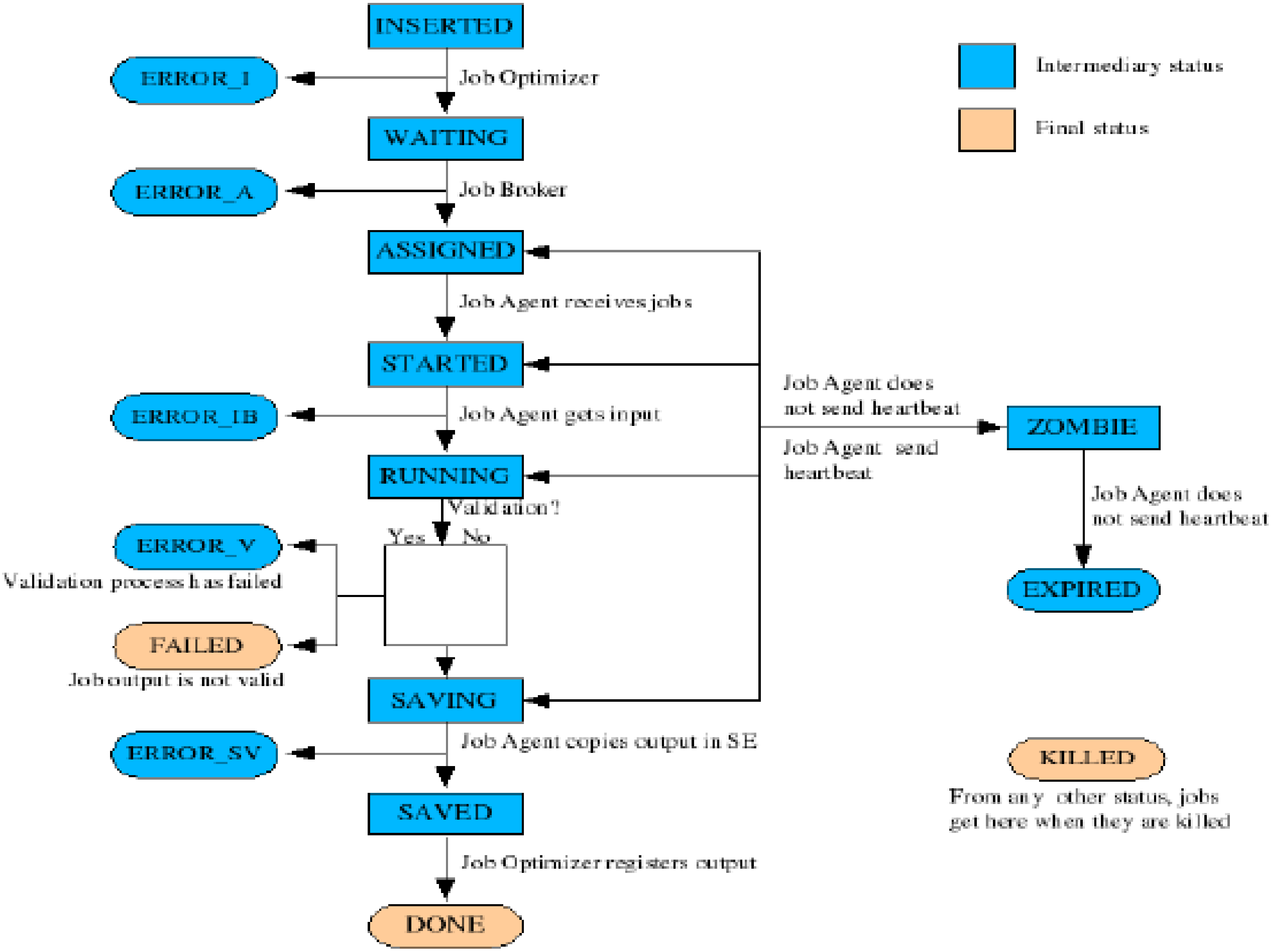
Batch analysis – JDL fields I

- **Executable:** Compulsory field where we give the lfn of the executable that should be stored in /bin or $V0/bin or $HOME/bin.
- **Arguments:** They will be passed to the executable.
- **Packages:** Type packages in the shell to see what kind of packages are installed.
- **InputFile:** The files that will be transported to the node where the job will run.
- **InputData:** It will require that the job will be executed in a site close to the files specified here.
- **InputDataList:** The filename in which the Job Agent will write the InputData list.
- **InputDataListFormat:** The format of the InputData list.

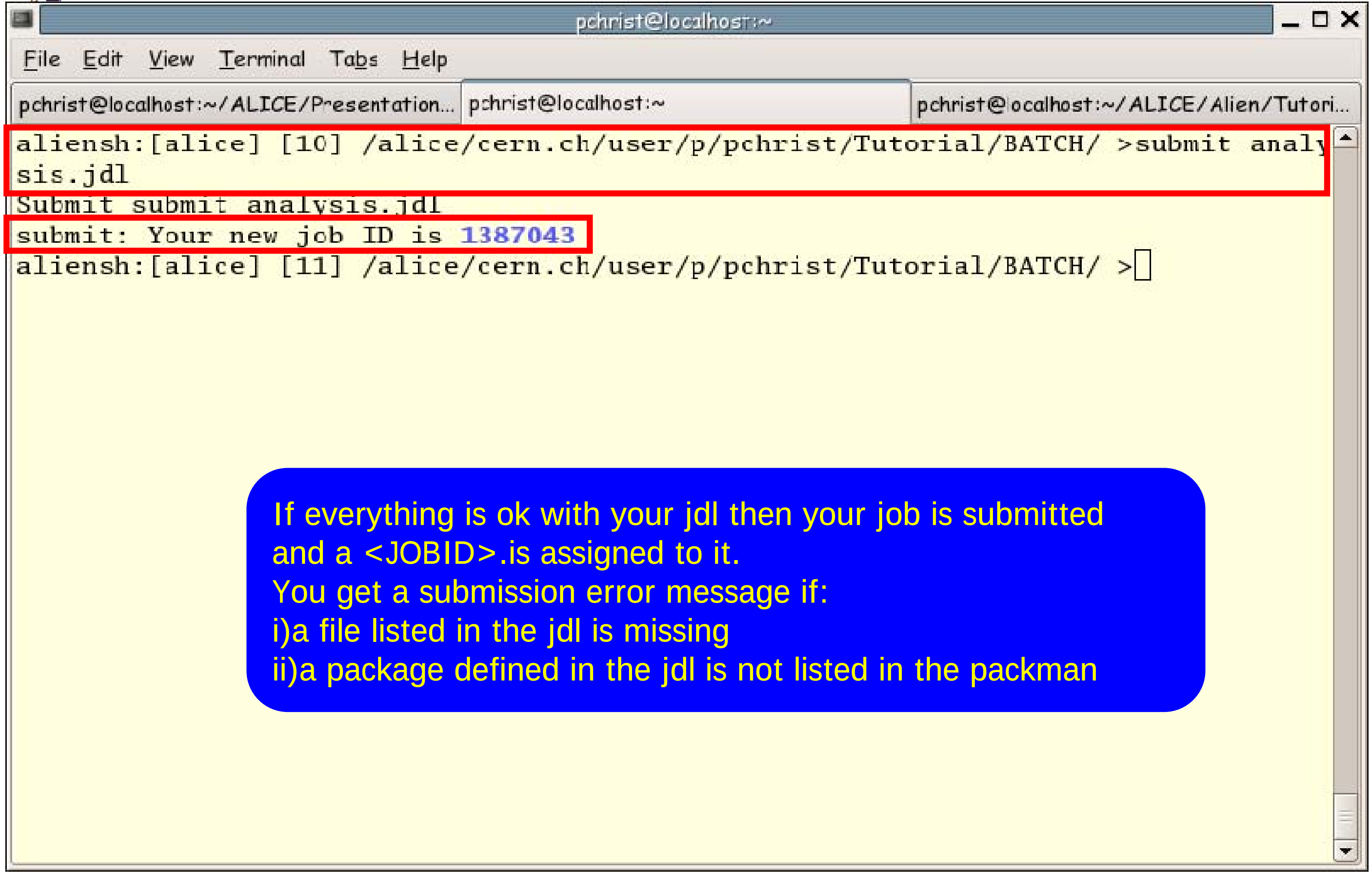
Batch analysis – JDL fields II

- **OutputFile:** The files that will be registered in the catalog once the job finishes.
- **OutputArchive:** What files will be archived in a zip file.
- **Validationcommand:** Specifies the script to be used as a validation script.
- **Email:** Receive a mail when the job finishes.
- **TTL:** The maximum run time of your job.
- **Split:** Split the jobs in several sub jobs.

Batch analysis – Job status



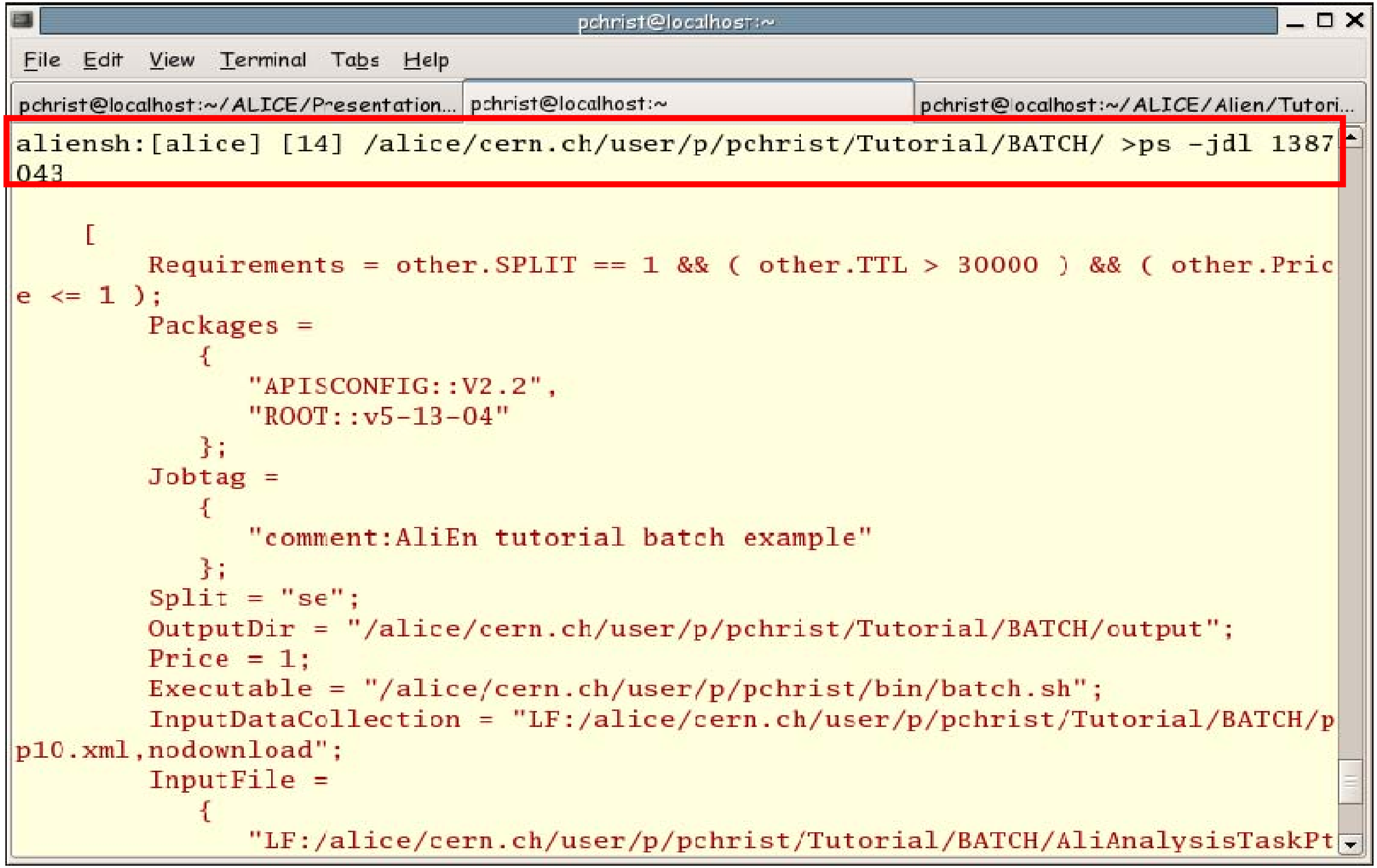
Batch analysis – Submitting jobs


pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
aliensh:[alice] [10] /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/ >submit analy
sis.jdl
Submit submit analysis.jdl
submit: Your new job ID is 1387043
aliensh:[alice] [11] /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/ >
```

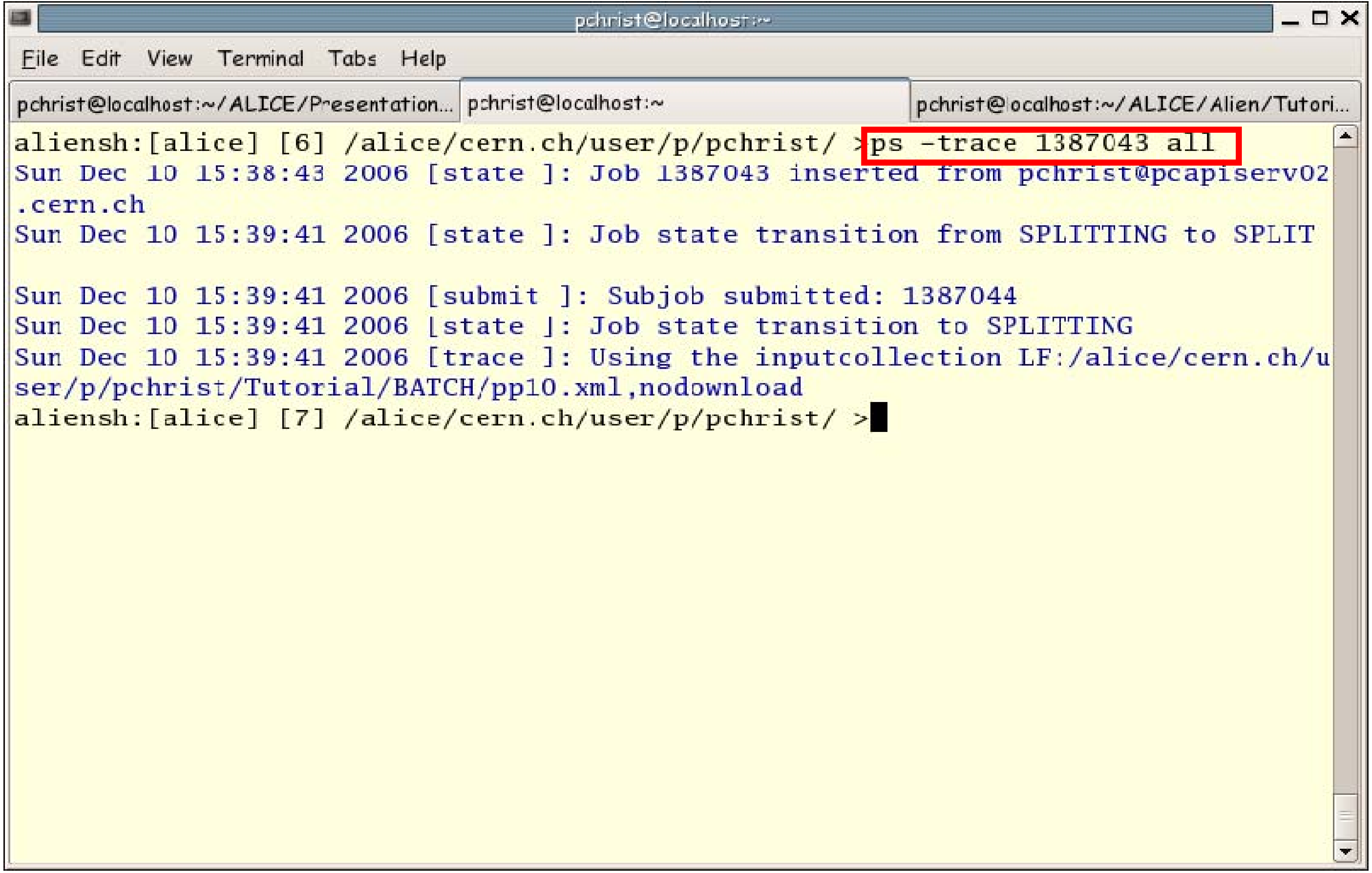
If everything is ok with your jdl then your job is submitted and a <JOBID>.is assigned to it.
You get a submission error message if:
i)a file listed in the jdl is missing
ii)a package defined in the jdl is not listed in the packman

Batch analysis – Checking the job status I


pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
aliensh:[alice] [14] /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/ >ps -jdl 1387
043


    [
        Requirements = other.SPLIT == 1 && ( other.TTL > 30000 ) && ( other.Pric
e <= 1 );
        Packages =
            {
                "APISCONFIG::V2.2",
                "ROOT::v5-13-04"
            };
        Jobtag =
            {
                "comment:AliEn tutorial batch example"
            };
        Split = "se";
        OutputDir = "/alice/cern.ch/user/p/pchrist/Tutorial/BATCH/output";
        Price = 1;
        Executable = "/alice/cern.ch/user/p/pchrist/bin/batch.sh";
        InputDataCollection = "LF:/alice/cern.ch/user/p/pchrist/Tutorial/BATCH/p
p10.xml,nodownload";
        InputFile =
            {
                "LF:/alice/cern.ch/user/p/pchrist/Tutorial/BATCH/AliAnalysisTaskPt
```

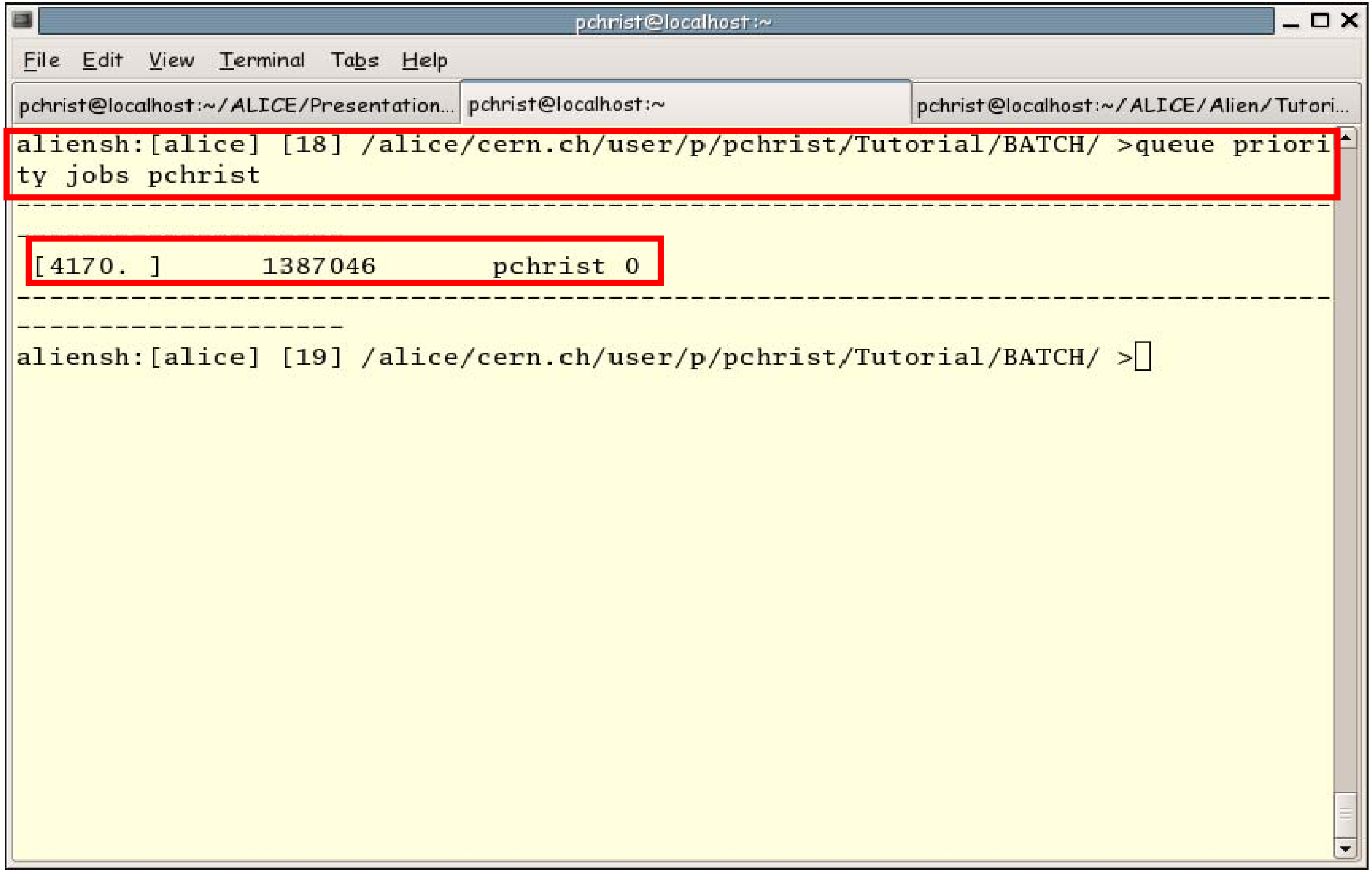
Batch analysis – Checking the job status II



pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
aliensh:[alice] [6] /alice/cern.ch/user/p/pchrist/ >ps -trace 1387043 all
Sun Dec 10 15:38:43 2006 [state ]: Job 1387043 inserted from pchrist@pcapiserv02
.cern.ch
Sun Dec 10 15:39:41 2006 [state ]: Job state transition from SPLITTING to SPLIT

Sun Dec 10 15:39:41 2006 [submit ]: Subjob submitted: 1387044
Sun Dec 10 15:39:41 2006 [state ]: Job state transition to SPLITTING
Sun Dec 10 15:39:41 2006 [trace ]: Using the inputcollection LF:/alice/cern.ch/u
ser/p/pchrist/Tutorial/BATCH/pp10.xml,nodownload
aliensh:[alice] [7] /alice/cern.ch/user/p/pchrist/ >
```

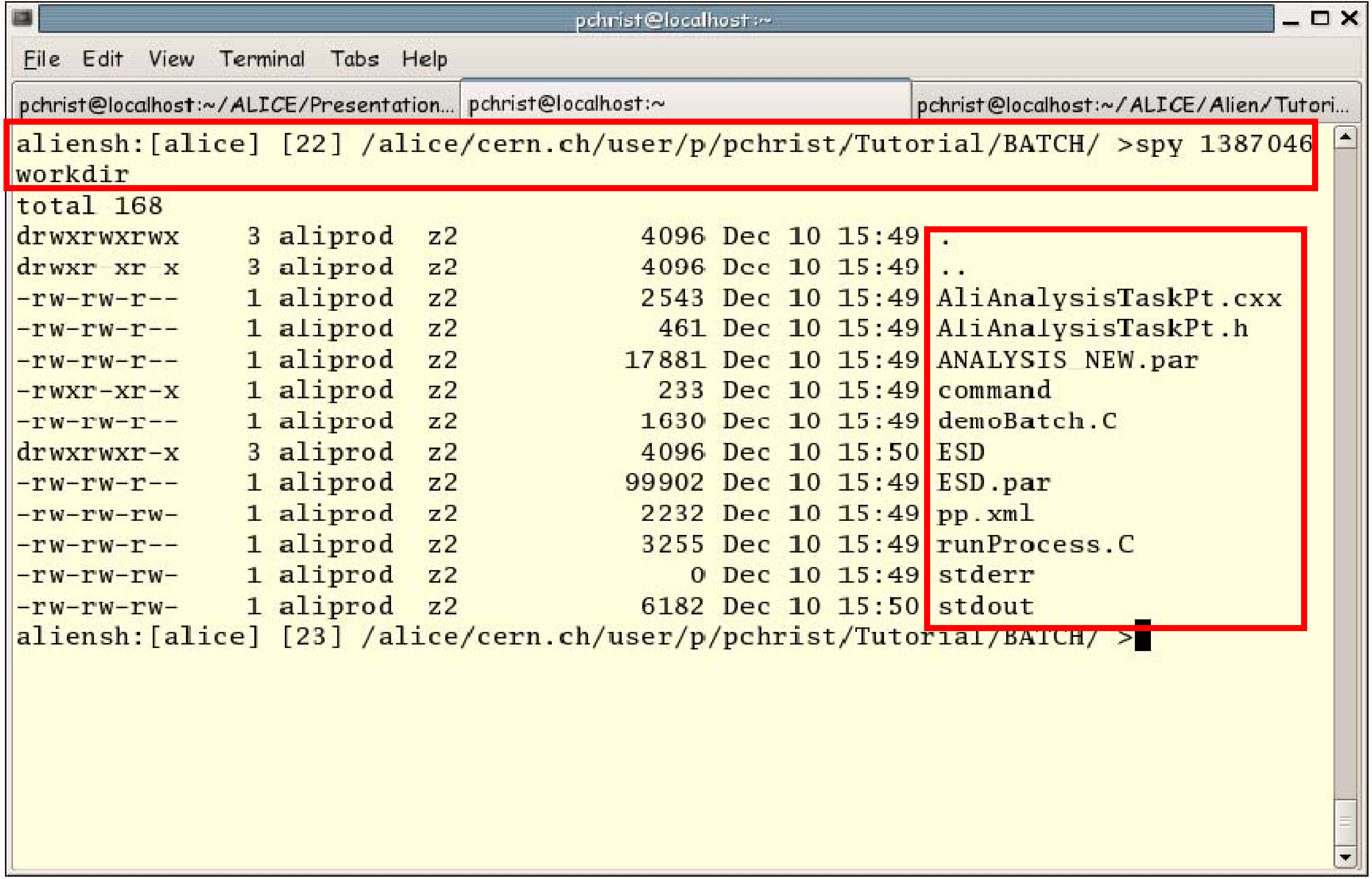
Batch analysis – Checking the priority



pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
aliensh:[alice] [18] /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/ >queue priori
ty jobs pchrist
--------------------------------------------------------------------------------
-
 [4170. ]      1387046       pchrist 0
--------------------------------------------------------------------------------
--------------------
aliensh:[alice] [19] /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/ >
```

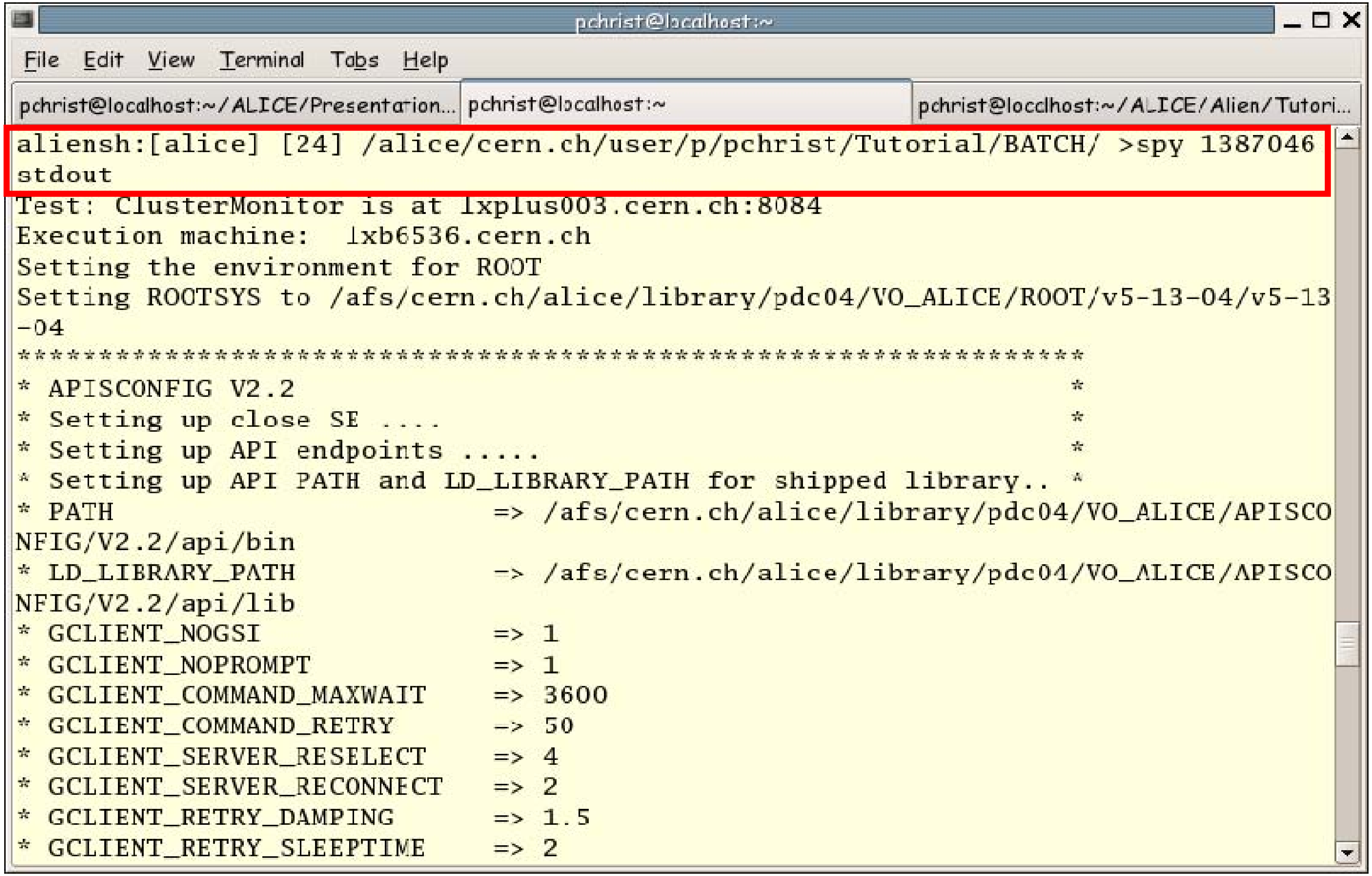
Batch analysis – Checking the job output I



pchrist@localhost:~
File Edit View Terminal Tabs Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
aliensh:[alice] [22] /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/ >spy 1387046
workdir
total 168
drwxrwxrwx    3 aliprod  z2          4096 Dec 10 15:49 .
drwxr xr x    3 aliprod  z2          4096 Dec 10 15:49 ..
-rw-rw-r--    1 aliprod  z2          2543 Dec 10 15:49 AliAnalysisTaskPt.cxx
-rw-rw-r--    1 aliprod  z2           461 Dec 10 15:49 AliAnalysisTaskPt.h
-rw-rw-r--    1 aliprod  z2         17881 Dec 10 15:49 ANALYSIS_NEW.par
-rwxr-xr-x    1 aliprod  z2           233 Dec 10 15:49 command
-rw-rw-r--    1 aliprod  z2          1630 Dec 10 15:49 demoBatch.C
drwxrwxr-x    3 aliprod  z2          4096 Dec 10 15:50 ESD
-rw-rw-r--    1 aliprod  z2         99902 Dec 10 15:49 ESD.par
-rw-rw-rw-    1 aliprod  z2          2232 Dec 10 15:49 pp.xml
-rw-rw-r--    1 aliprod  z2          3255 Dec 10 15:49 runProcess.C
-rw-rw-rw-    1 aliprod  z2             0 Dec 10 15:49 stderr
-rw-rw-rw-    1 aliprod  z2          6182 Dec 10 15:50 stdout
aliensh:[alice] [23] /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/ >
```

Batch analysis – Checking the job output II


pchrist@localhost:~
File
Edit
View
Terminal
Tabs
Help
pchrist@localhost:~/ALICE/Presentation...
pchrist@localhost:~
pchrist@localhost:~/ALICE/Alien/Tutori...

```
aliensh:[alice] [24] /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/ >spy 1387046
stdout
Test: ClusterMonitor is at lxplus003.cern.ch:8084
Execution machine:  lxb6536.cern.ch
Setting the environment for ROOT
Setting ROOTSYS to /afs/cern.ch/alice/library/pdc04/VO_ALICE/ROOT/v5-13-04/v5-13
-04
*****************************************************************
* APISCONFIG V2.2                                               *
* Setting up close SE ....                                      *
* Setting up API endpoints .....                                *
* Setting up API PATH and LD_LIBRARY_PATH for shipped library.. *
* PATH                        => /afs/cern.ch/alice/library/pdc04/VO_ALICE/APISCO
NFIG/V2.2/api/bin
* LD_LIBRARY_PATH             => /afs/cern.ch/alice/library/pdc04/VO_ALICE/APISCO
NFIG/V2.2/api/lib
* GCLIENT_NOGSI               => 1
* GCLIENT_NOPROMPT            => 1
* GCLIENT_COMMAND_MAXWAIT     => 3600
* GCLIENT_COMMAND_RETRY       => 50
* GCLIENT_SERVER_RESELECT     => 4
* GCLIENT_SERVER_RECONNECT    => 2
* GCLIENT_RETRY_DAMPING       => 1.5
* GCLIENT_RETRY_SLEEPTIME     => 2
```

Batch analysis – Merging the output

- The output of every splited job is listed under:
  - /proc/$username/$JOBID if the OutputDir is not defined in the jdl.
  - OutputDir if it is defined in the jdl.
- In order to merge the several output files you have to run a post process:
  - You can find the macro that deals with this under /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/histomerge.C
  - [bash]$ alienroot
  - root [0] .L histomerge.C
  - root [1] histomerge($OutputDirPath,$pattern,$mergefile)

- OutputDir: is the output directory of the master job (jdl)
- Pattern: the zip file that you create in your selector
- Mergefile: the desired locally stored merge file

Batch analysis – Try it out I

- Copy the following files to $HOME/AliEn/Batch:
  - /alice/cern.ch/user/p/pchrist/Tutorial/INTERACTIVE/tag.xml
  - /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/ANALYSIS_NEW.par
  - /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/AliAnalysisTaskPt.h
  - /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/AliAnalysisTaskPt.cxx
  - /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/runProcess.C
  - /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/demoBatch.C
  - /alice/cern.ch/user/p/pchrist/bin/batch.sh
  - /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/analysis.jdl
  - /alice/cern.ch/user/p/pchrist/Tutorial/BATCH/CreateXML.C

Batch analysis – Try it out II

- Open the runProcess.C and uncomment the lines needed for the creation of the new xml collection.
- Run the macro with alienroot and open your global.xml that will be created locally.
- Open the analysis.jdl and modify the Email, InputFile, InputDataCollection and OutputDir fields.
- Copy the local batch.sh to your AliEn $HOME/bin.
- Copy the local analysis.jdl to your AliEn $HOME/Tutorial/XML/jdl.

Batch analysis – Try it out III

- Copy the local global.xml to your AliEn $HOME/Tutorial/XML/xml.
- Copy the local *.par to your AliEn $HOME/Tutorial/XML/par.
- Copy the local AliAnalysisTaskPt.* to your AliEn $HOME/Tutorial/XML/selectors.
- Copy the local demoBatch.C and runProcess.C to your AliEn $HOME/Tutorial/XML/macros.
- Go to your AliEn $HOME/Tutorial/XML/jdl and submit the job by typing: “submit analysis.jdl”.

Batch analysis – Try it out IV

- Check your job priority by typing: “queue priority jobs $username”.
- Display the jdl of your job by typing: “ps -jdl $jobid”.
- Trace the status of your job by typing: “ps -trace $jobid”.
- When the job 's status turns to RUNNING you can get the stdout and stderr of the job by typing:
  - “spy $jobid stdout”.
  - “spy $jobid stderr”.
- Once the job is finished, merge the output and store it locally as Pt.Merged.root

References I

- Registration – Certificates:
  - http://alien.cern.ch/twiki/bin/view/Alice/UserRegistration
  - https://ca.cern.ch/ca/
- AliEn:
  - http://alien.cern.ch
- Gshell:
  - http://alien.cern.ch/twiki/bin/view/AliEn/GAPI
- User's guide:
  - http://project-arda-dev.web.cern.ch/project-arda-dev/alice/apiservice/AA-UserGuide-0.0m.pdf

References II

- ***aliensh*** Grid Command Online Reference V1.0
  - http://project-arda-dev.web.cern.ch/project-arda-dev/alice/apiservice/guide/guide-1.0.htm
- Previous tutorials:
  - http://aliceinfo.cern.ch/Offline/Analysis/Tutorial/
- Event Tag System:
  - http://pcaliweb02.cern.ch/Offline/Analysis/RunEventTagSystem/EventTags.html#Event%20tag%20system
  - https://edms.cern.ch/document/788315/1 **(INTERNAL NOTE)**

References III

- Creation of tag files:
  - http://pcaliweb02.cern.ch/Offline/Analysis/RunEventTagSystem/EventTagsCreation.html#Create%20tags%20howto
- Analysis using the Event Tag System:
  - http://pcaliweb02.cern.ch/Offline/Analysis/RunEventTagSystem/EventTagsAnalysis.html#Analysis%20with%20tags

References IV

- File catalog structure – Queries:
  - http://pcaliweb02.cern.ch/Offline/Analysis/RunEventTagSystem/RunTags.html#Run/File%20metadata
- File level metadata:
  - http://cern.ch/Oldenburg/MetaData/MetaData.doc
- Analysis framework
  - http://indico.cern.ch/materialDisplay.py?contribId=19&amp;sessionId=3&amp;materialId=slides&amp;confId=a056304

Analysis framework

Andrei Gheata

ALICE offline week, 5 October ‘06

Purpose

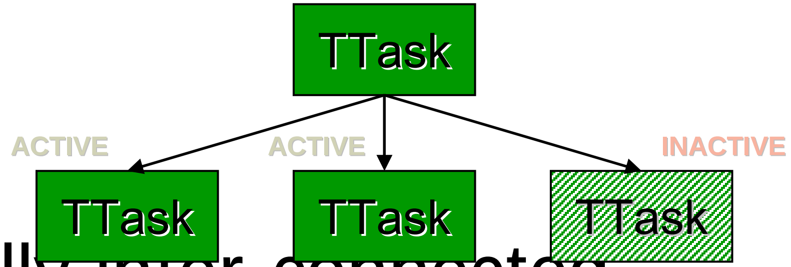
- Provide easy-to-use tools to allow data analysis in a coherent way
- Suitable for analysis ranging from simple to very complex tasks in a distributed environment
- Allow splitting complex analysis tasks in independent functional blocks possibly usable by other analysis

Functionality

- Basic ideas described at the last offline week
- Data-oriented model composed of independent tasks
  - Task execution triggered by data readiness
- Parallel execution and event loop done via ***TSelector*** functionality
- Analysis execution performed on event-by-event basis

Structure

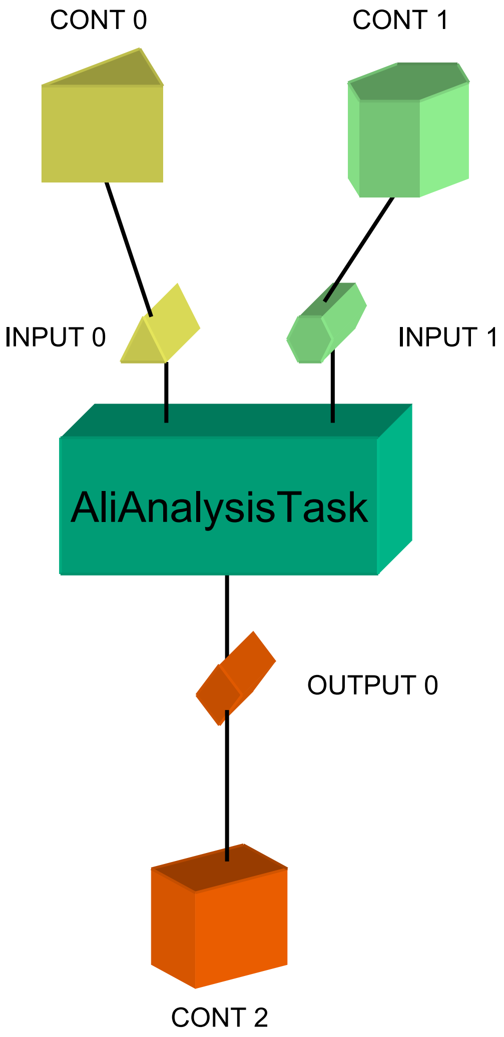
- Analysis <u>may</u> be split in functional modules
  - At least one
  - Deriving from TTask



- Modules are not manually inter-connected
  - Connected just to input/output data containers
  - A data container has one provider and possibly several clients
  - A module becomes active when all input data is ready

Data-oriented model

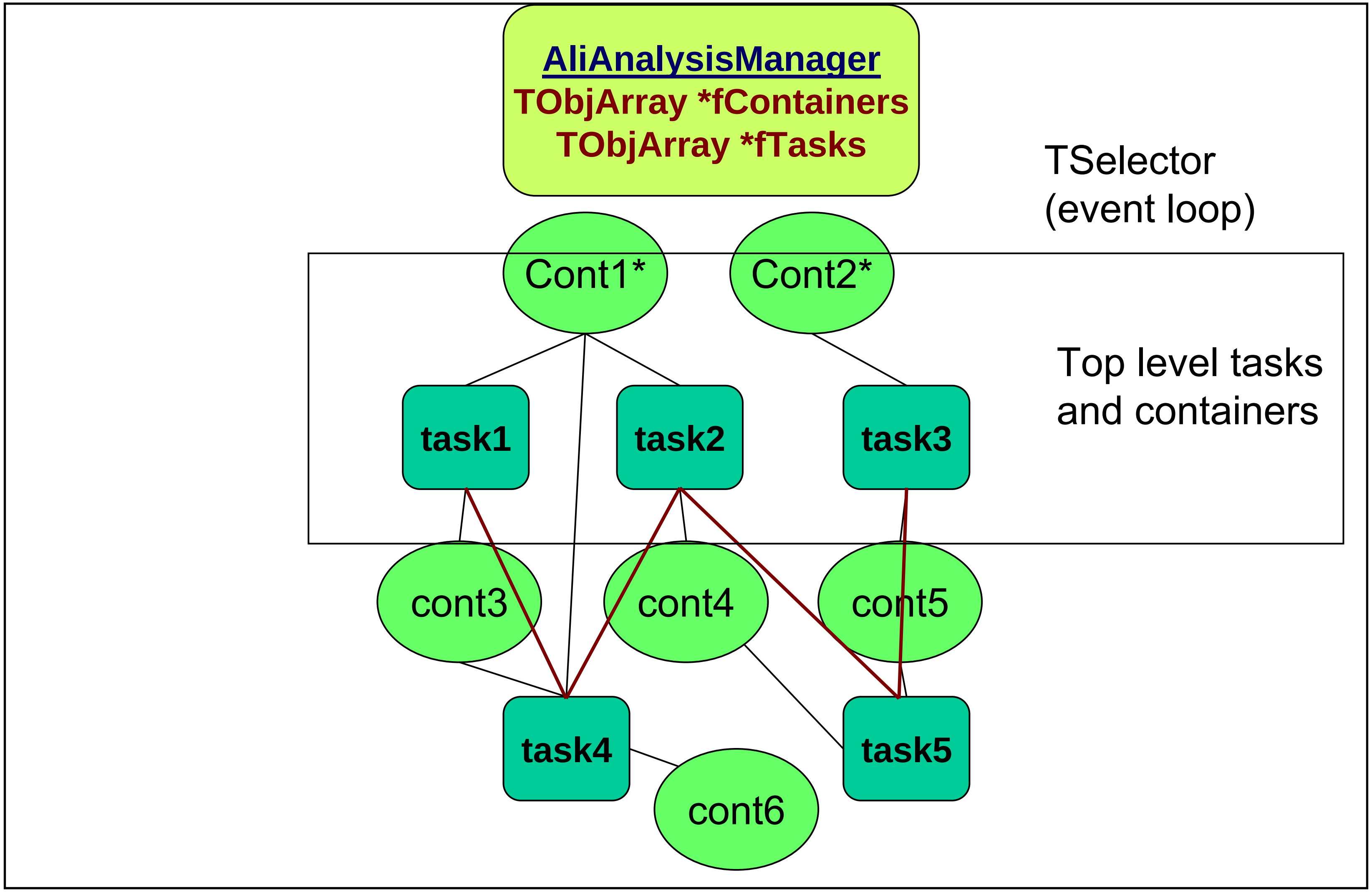
- Data type formalized by ***TClass*** usage
- Any module declares a number of input data slots
  - Each slot must be connected to a data container of the corresponding type at run time
- Modules provide data at one or more output slots



Management

- Analysis modules managed by a ***TSelector***-derived class
  - Provides access to <u>initial input data</u> (ESD's, kinematics, whatever...) for <u>the top-level containers</u>
  - Initiates the main event loop over the entries of the input trees, calling the Exec() method for the <u>top-level tasks</u>
- Input data is generally a TChain, but the framework can manage other data types
  - Retreival by event tags mechanism (see talk from Panos) – to be interfaced
- Parallelizing analysis execution
  - Functionality provided by TSelector@PROOF (see talk from Jan Fiete)

Data flow structure



Implementation

- Code in AliRoot
  - Inside **ANALYSIS** module
  - Classes: AliAnalysysManager, AliAnalysisTask, AliAnalysisDataContainer, AliAnalysisDataSlot, AliAnalysisContainerRL
  - Besides the last class, no dependency to AliRoot
- Separate library to be loaded
  - libANALYSIS_NEW
- Demo for package usage: testAna.C inside ANALYSIS folder

AliAnalysisManager : public TSelector

- CreateContainer(const char *name, TClass *data_type, EAliAnalysisContType cont_type)
  - Mandatory to define all data containers that will assembly the analysis
  - Container types:
    - kInputContainer – minimum 1 input container needed
    - kNormalContainer – containers used for communication between task modules
    - kOutputContainer – minimum 1 output container

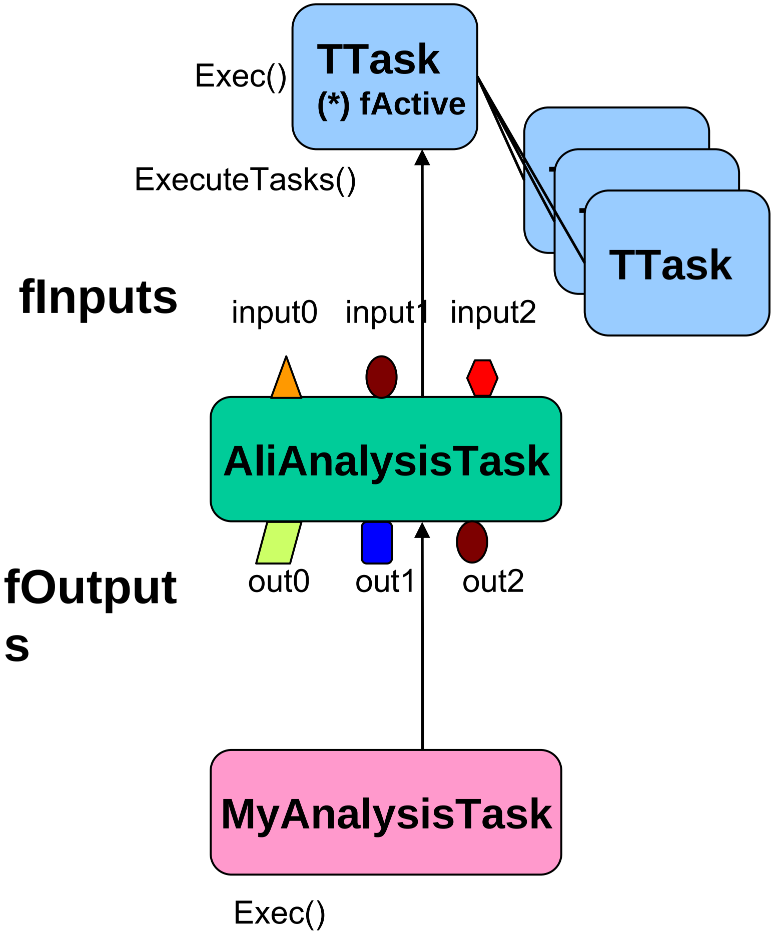
AliAnalysisManager (continued)

- AddTask(AliAnalysisTask *task)
  - At least 1 task per analysis (top task)
- ConnectInput(pTask, islot, pContainer)
- ConnectOutput(pTask, islot, pContainer)
  - Mandatory for all data slots defined by used analysis modules
- InitAnalysis()
  - Performs a check for data type cosistency and signal any illegal circular dependencies between modules
  - To be called by TSelector::Init()
- ExecAnalysis()
  - Starts the analysis
  - To be called by TSelector::Process()

AliAnalysisTask : public TTask

- User analysis module MUST subclass this
- DefineInput/Output(Int_t islot, TClass *type)
  - Mandatory at least 1 input & 1 output
  - Usually declared in the class constructor
- virtual void Exec(Option_t *option) = 0
  - Manadatory to implement in the derived class
  - This actually implements how the analysis module processes input data

Analysis module (task)



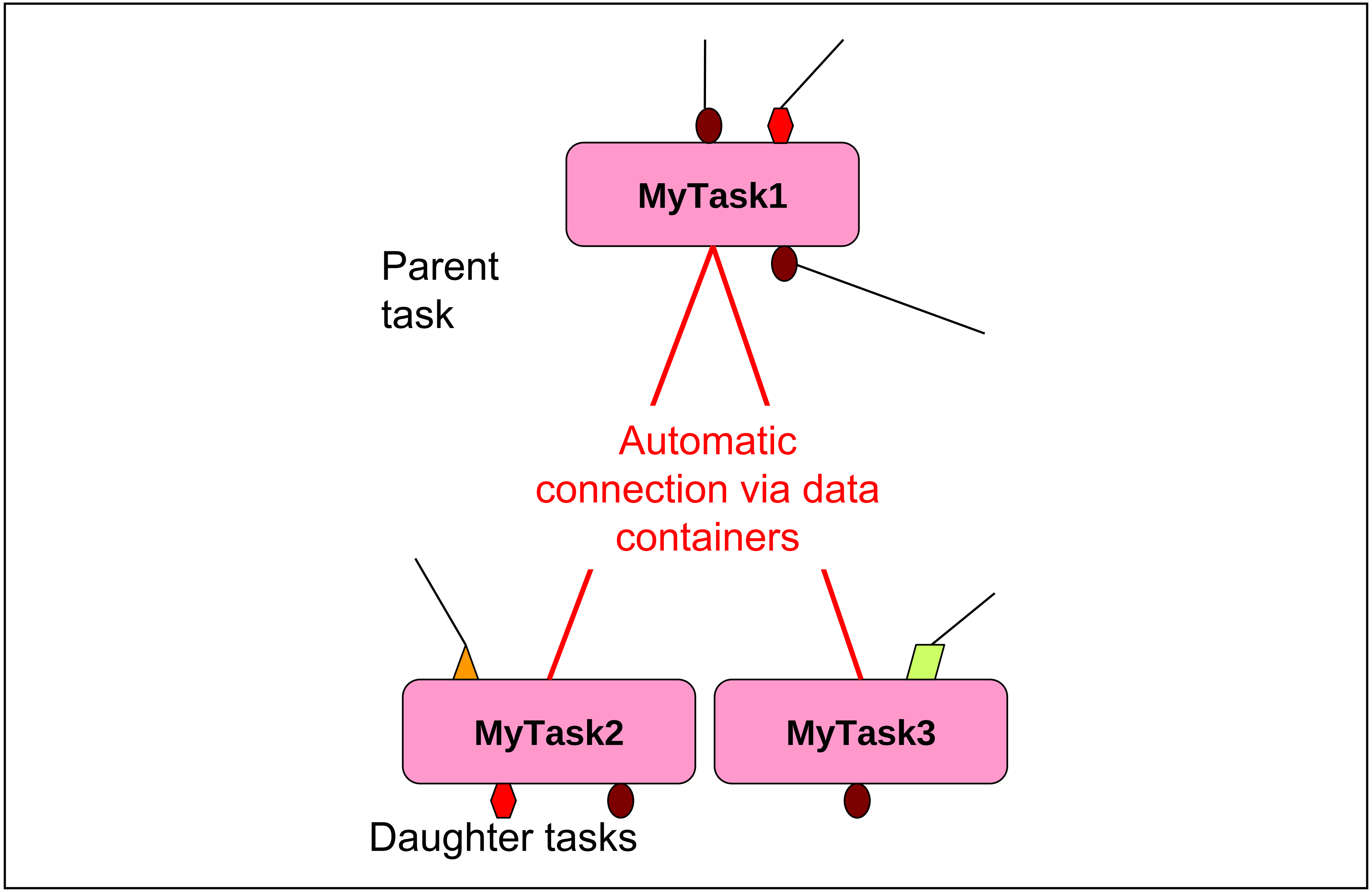
How to implement Exec()

- Accesing data from input slots
  - When Exec() is called, data will be always available at all declared inputs
  - Use: MyClass *data = (MyClass*)GetInputData(islot)
- Processing input data
  - In case of events, organize track loop
- Publishing the result at output
  - Mandatory to be done at the end of event processing
  - Use: PostData(Int_t islot, TObject *result, Option_t *option)
    - Will notify the container connected to output and all dependent daughter tasks that data is ready
    - Subtasks activated when all inputs are ready, executed by the last provider
    - Option – specifies if data should be written to a file

AliAnalysisDataContainer

- Normally a class to be used ‘as is’
  - Enforcing a data type deriving from TObject
  - For non-TObject (e.g. basic) types one can subclass and append the needed types as data members
- Three types of data containers
  - Input – containing input data provided by AliAnalysisManager
  - Transient – containing data transmitted between modules
  - Output – containing final output data of an analysis chain, eventually written to files.
- One can set a file name if the content is to be written
- ***AliAnalysisContainerRL*** – special container using ***AliRunLoader*** to access specific data
  - To be moved in a separate library

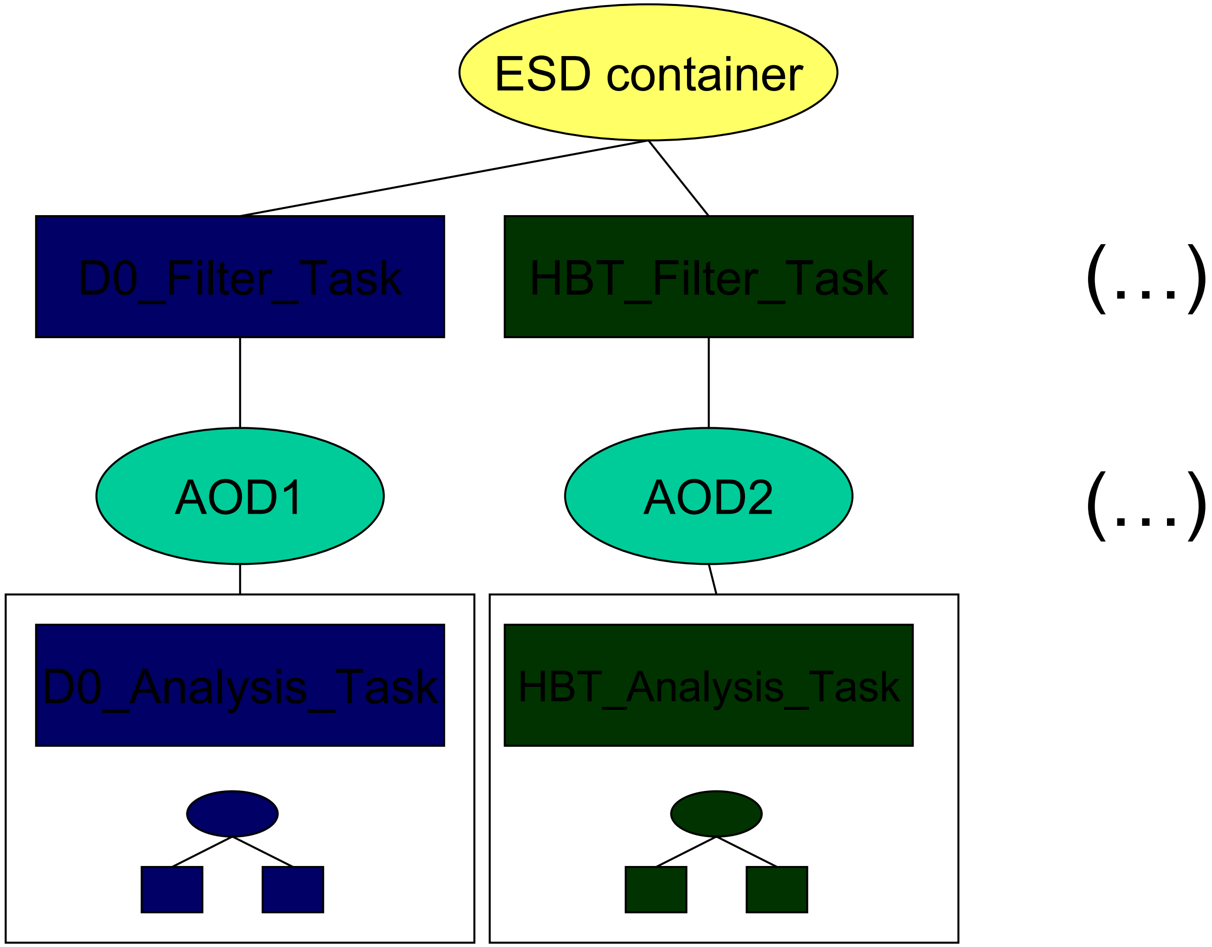
Connection via data containers



AliAnalysisDataSlot

- Input/Output task slots
- Not a class to be handled by users
  - Can be declared/created in association with a task, using methods belonging to ***AliAnalysisTask***

Example: AOD parallel production



Conclusions

- Analysis framework in AliRoot
  - Provides all needed functionality, but there are also some basic to-do's left
    - Connection to event tag mechanism
    - TSelector functionality connection
- Framework quite flexible and simple to use
  - See ANALYSIS/testAna.C macro as a simple example on how to use the framework
- Additional functionality, bug fixes, optimizations certainly needed
  - Feedback would help