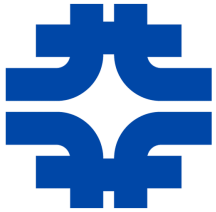

FNAL Software School Introduction

Matt Herndon,

University of Wisconsin – Madison

FNAL Software School

- FNAL Software School
  - Programming Reconstruction Software for Large Computing Projects
  - HEP Software training
- Goal
  - Learn how to write well designed and effective, reconstruction software that integrates well into a large scale computing project
- Project(s)
  - Tracking reconstruction is used as an example of a typical reconstruction software project in particle physics
  - Active participation is a central element of the school!
  - You will pursue several small reconstruction software development projects chosen to illustrate problems and techniques in software engineering for large computing projects

Personnel

- Lead instructor and course developer
  - Matthew Herndon
  - I am an expert in tracking reconstruction software having led the CDF tracking group and programed reconstruction software for a silicon strip tracker and muon systems at CDF and CMS
- TAs and developers
  - Our team of TA's has extensive experience in software engineering and reconstruction in particle physics at Fermilab
  - A core group advised and contributed to the development of the course software
  - Robert Kutschke, Marc Paterno, Christopher Jones, Patrick Gartung, Elizabeth Sexton-Kennedy, Sudhir Malik, Mike Hildreth, Qiming Lu, James Kowalkowski, William Tanenbaum , Paul Russo, Christopher Green, William D Dagenhart, William M Tanenbaum
  - They will assist you in the daily software projects

Resources

- Course indico page:
  - https://indico.fnal.gov/conferenceDisplay.py?ovw=True&confId=8568
  - Introduction, timetable and slides
- Course Redmine page:
  - https://cdcvs.fnal.gov/redmine/projects/fnal-soft-school-summer-2014
  - Overview page
  - Front Wiki page with introduction primary resources
    - Repository, course syllabus and exercises, best practices guide
  - Daily pages with detailed instructions, exercises, and resources
  - Issue tracker – requires logging into redmine using your FNAL Services account user name and password.
- Course software repository:
  - https://github.com/herndon/FNALComp/tree/production
- Doxygen Code Browser
  - http://lpc.fnal.gov/FNALsoftwareSchool/CodeBrowser/index.html
- Contact information
  - Instructor and Ta's: listserve fnal_softwareschool2014@fnal.gov
  - Software and technical issues can be reported using the redmine issue tracker

Today's Activities

- Introductory slides
- Review Day 0 pre-course exercise
  - Make sure you can access, build and run the course software.
- Lecture
  - Course code infrastructure
  - Programming in a large software project
  - Hit reconstruction
  - On data/algorithm abstraction
- Daily project: Cluster finding and hit reconstruction
  - Performed in groups of 3
  - Planning: jointly with TAs
  - (Lunch)
  - Programming time
  - Progress assessment
  - Programming time
  - Assessment and discussion time, informal

Course Goals

- Goal
  - Learn how to write well designed and effective reconstruction software that integrates well into a large scale computing project
- What does that mean?
  - Follows best practices
    - Many of the best practices are there to facilitate the elements of the goal.
  - Easy to read
    - A user or other developer can read and understand quickly what your code does.
  - Easy to maintain
    - Need to improve something? Well designed code will often let you do so with a change at a single point without effecting any of the classes and functions that use the code you've changed.
  - Simple
    - The simplest solution is used when various solutions are equally effective.
  - Safe
    - Data elements are safe from being altered when they should not be.
  - Fast uses minimal memory
    - A fact of particle physics computing is that we deal with large data sets and are CPU and memory limited.
  - Effective
    - Defined in terms of the project goal. In reconstruction typically, efficient, accurate, and low fake rate (reconstruction of Hits, Tracks that don't exist!) reconstruction of objects.

Day 0 exercise review

- Day 0 exercise
  - Setup accounts,
  - Download software
  - Build and run
  - Modify software accessing each of the generated data objects to make histograms
- If building and running the software did not work we should address the issue now.
- The day 0 exercise is designed to assess whether you have the necessary level of knowledge and skills to be in the course.
- We expect
  - Ability to do basic programming in C++
    - For instance you have written analysis software
  - Familiarity with Unix systems such that the scripts we have provided make sense.