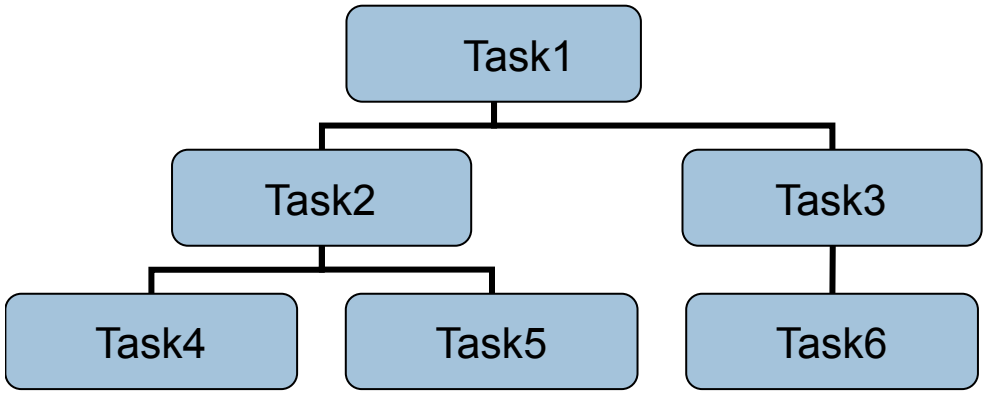
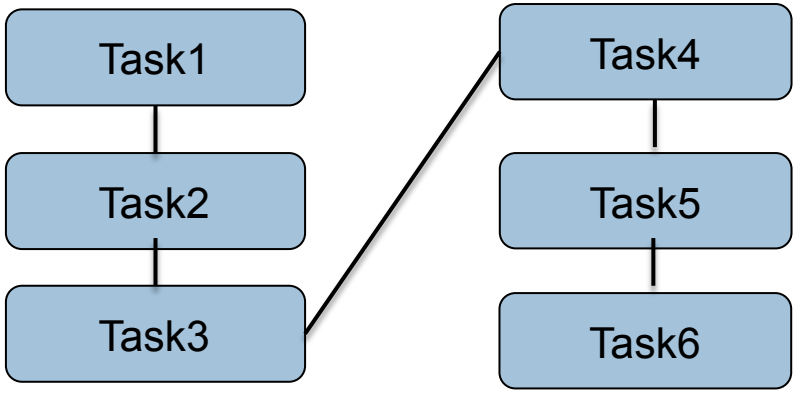
FAIRROOT RECONSTRUCTION

Panda Computing Week 2012, Torino

Everything is a task

- A task is a procedure to read an already existing data level and produce a new one, or change the existing one.
- Complex structures of Tasks and Subtasks are possible
  - Up to now only a simple sequence of tasks is used





Hands On: Create a dummy task

- create a digitization directory in torinoDetector
- Copy content of the template/NewTask directory into new digitization directory
- Run the rename script
- Try to understand the new Class
- Edit files and try to compile PandaRoot

Basic functions of a task

- Init()
  - Do all the initialization which are needed to run the task
    - Get pointer to the data classes
    - Initialize variables
    - Whatever you have to do before starting the event loop
- ReInit()
  - Do the same if run conditions (Unique run ID) change
- SetParContainers()
  - Define the parameter container which are needed by the task
- Exec()
  - Here the work is done (whatever this might be)
  - The function is called for each event
- Finish()
  - The function is called at the end of the run
  - Here you should do all the cleanup

Reconstruction Macro

- Load the necessary libraries

```
void run_digi()
{
  gROOT->LoadMacro("$VMCWORKDIR/gconfig/rootlogon.C");
  rootlogon();
  gSystem->Load("libPndTorinoDetector.so");
```

- Define Input and Output

```
  // Input file (MC events)
  TString inFile = "data/testrun.root";
  // Parameter file
  TString parFile = "data/testparams.root";
  // Output file
  TString outFile = "data/testhits..root";

  FairRunAna *fRun= new FairRunAna();
  fRun->SetInputFile(inFile);
  fRun->SetOutputFile(outFile);
```

Reconstruction Macro

Setup of the Runtime Database

```
FairRuntimeDb* rtdb = fRun->GetRuntimeDb();
 FairParRootFileIo* parInput1 = new FairParRootFileIo();
 parInput1->open(parFile.Data());
 rtdb->setFirstInput(parInput1);
```

Define the task

```
PndTorinoDetectorHitProducerSmearing* hitProducer =
      new PndTorinoDetectorHitProducerSmearing();
 fRun->AddTask(hitProducer);
```

Initialize and run the reconstruction

```
fRun->Init();
 timer.Start();
 fRun->Run();
```

Hands On: Create and run the macro

- Create the simple macro from the slides before
- Run the macro
- Look for the output on the screen