



Introduction to the Jlab Computing Environment

Petr Stepanov

petrs@jlab.org

June 17, 2021

Presentation Outline

- Secure Shell Protocol (SSH)
- Environment Variables
- Utilizing the JLab Computing Environment
- Quick Introduction to the Geant4
- Running Geant4 Example Program on the Farm
- Using ROOT to Analyze Geant4 Output Data (Next Week)

Secure Shell Protocol (SSH)

– A secure connection to another computer to perform *remote executing* of the shell commands. Authentication on remote computer works via password or key file.



Your data passes through many networks. It can be intercepted and modified.

Installation of SSH Software

SSH software needs to be running on the remote machine and SSH client on your local machine to make a connection.

On Mac and Linux SSH client *installed by default*. On Windows use [WSL](#) or [PuTTY](#).

SSH Command Syntax

```
ssh -Y username@serverhost
```

– `username` is the user name on the remote computer (not your local user);

– `serverhost` can be either IP address or domain name of the remote machine.

– `-Y` argument forwards the X11 (X server, X.Org) over the SSH

Environment Variables

– A set of variables (NAME=VALUE pairs) that affect how your programs behave on computer. By convention, variable names are capitalized.

Listing the Environment Variables

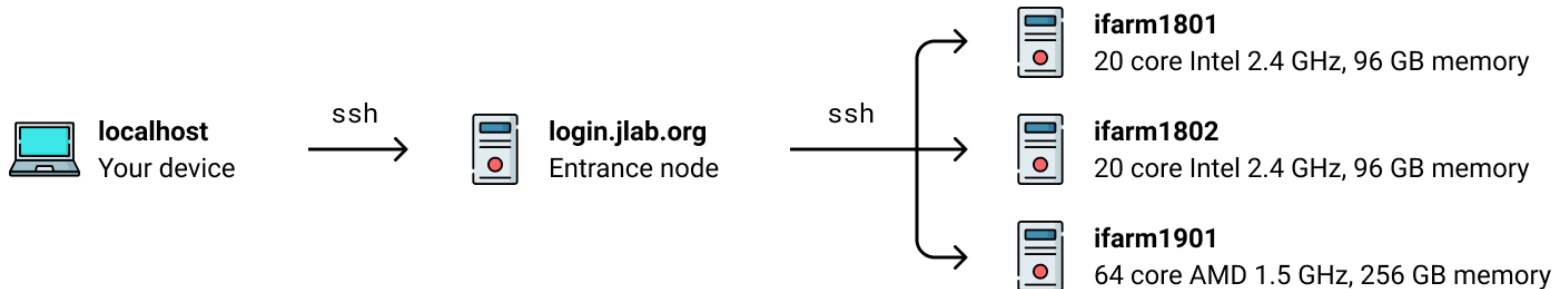
- On Mac and Linux use `env` command in Terminal.
To output value of a single variable use `echo $VARIABLE_NAME` (\$ symbol substitutes variable name with variable value).
- Windows also has environment variables. To list them type `env:` in the PowerShell.

Few Environment Variables Worth Knowing

<code>USER=petrstepanov</code>	- name of the currently logged in user
<code>HOME=/home/petrstepanov</code>	- home folder location of the current user
<code>LANG=en_US.UTF-8</code>	- locale (number format, currency, paper size...)
<code>PATH=/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin...</code>	- ordered list of paths that Linux will search for executables when running a command.
<code>LD_LIBRARY_PATH=...</code> <code>DYLD_LIBRARY_PATH</code> on macOS	- shared libraries lookup paths; these libraries are dynamically linked at the executable runtime.

Accessing the JLab Computing Environment

To connect to the Computing Environment one needs to obtain a JLab username and password from the [Computer Sciences Helpdesk](#).



Accessing Pre-Compiled ROOT and Geant4 Executables

System paths to the pre-compiled Geant4 and ROOT executables are appended to the “default” user’s environment variable values. Therefore executables become accessible.

List available environment versions: `source /site/12gev_phys/softenv.csh`

Source specific version of environment: `source /site/12gev_phys/softenv.csh 2.4`

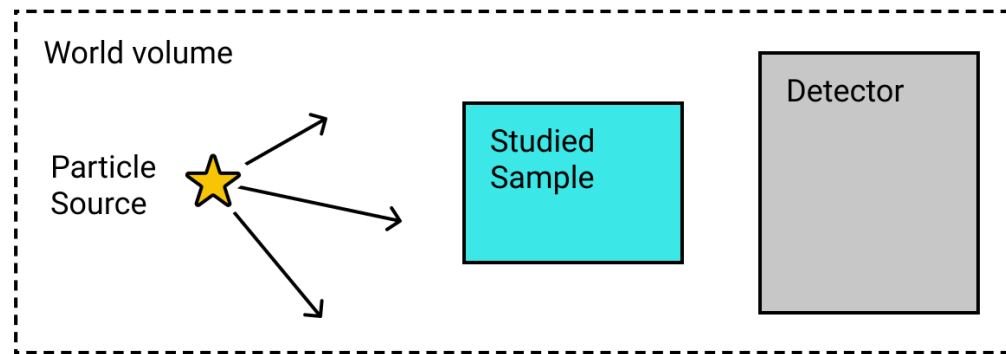
OPTIONAL: Utilizing the Multi-Threading (MT) Mode for Geant4

Append the `MT` suffix to the `GEANT4_VERSION` environment variable and re-source the environment. Somewhat impolite: a lot of folks are sharing Farm nodes at the same.

```
set GEANT4_VERSION=${GEANT4_VERSION}MT
source /site/12gev_phys/softenv.csh 2.4
```

GEANT4 Framework Basics

GEometry ANd Tracking. A toolkit for the simulation of the passage of particles through matter. Applied in HEP (nuclear, accelerator physics), medical science (tumors), space.



“Building blocks” of a trivial GEANT4 simulation.

- World volume.** ”Mother” volume that contains other volumes in the simulation.
- Particle source.** Defines size, direction, energy (spectrum), kind of emitted particles.
- Studied sample.** Set of geometrical shapes (solids) with defined elemental composition (logical volume) that are positioned in space (physical volume).
- Detector.** Sensitive volume that registers physical quantities: energy deposition, dose, track length, and more...

Refer to the [GEANT4 YouTube Mini Course](#) by Jing Liu to learn more.

Running a GEANT4 Example

GEANT4 has “basic”, “extended” and “advanced” sets of examples that provide best practices and conventions of writing your own GEANT4-based program. Find the *example documentation* and *source code* on [Geant4 GitHub repository](#).

Connecting to the Farm and Sourcing the Environment

```
ssh -Y <your-username>@login.jlab.org
ssh ifarm # connects to a random node
source /site/12gev_phys/softenv.csh 2.4
```

Obtaining out the GEANT4 examples source code

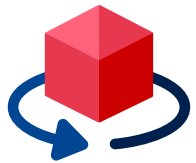
```
cp -r -v
/u/site/12gev_phys/2.4/Linux_CentOS7.7.1908-gcc9.2.0/geant4/4.10.06.p02/
source/geant4.10.06.p02/examples ~/examples
```

Building the example

```
cd ~/examples/basic/B1 # navigate into an example folder
mkdir build && cd build # create a sub-directory for the build
cmake -DGeant4_DIR=$G4LIB/Geant4-$G4DATA_VERSION/ ../.. # detect libraries and generate GNU Makefile
make # compile sources and link the executable
make install # (optional) install in usr/local/bin
```

Interactive and Batch GEANT4 Modes

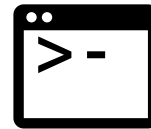
A program compiled with GEANT4 toolkit can be executed in “interactive” and “batch” modes.



Interactive Mode

Run few events for visual validation of the geometry.
Resource hungry – OpenGL.

```
./<executable>
```



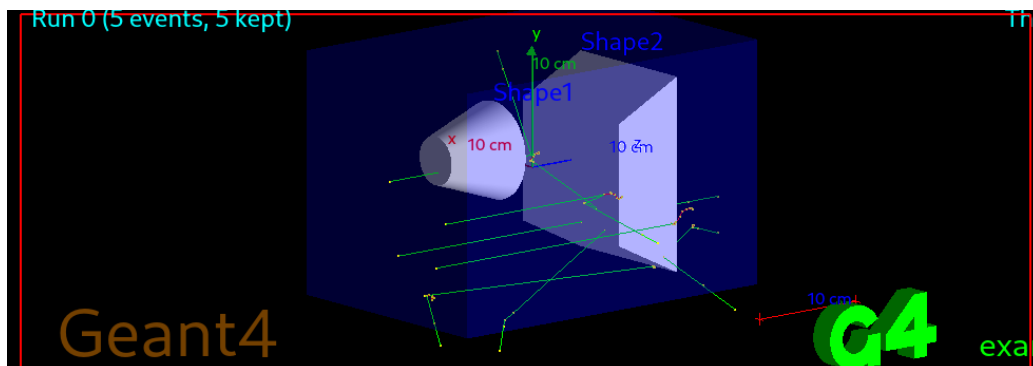
Batch Mode

Run many events and obtain Monte-Carlo numerical results.

```
./<executable> <macro-file>
```

Running example different modes

Documentation on running any Geant4 example can be found in corresponding example’s README file in GitHub repository. For exampleB1 [run instructions are here](#).



Visualization of the 5 events of 6 MeV gamma rays simulated in the GEANT4 exampleB1.

Neutral particles

Negatively charged particles

Positively charged functions

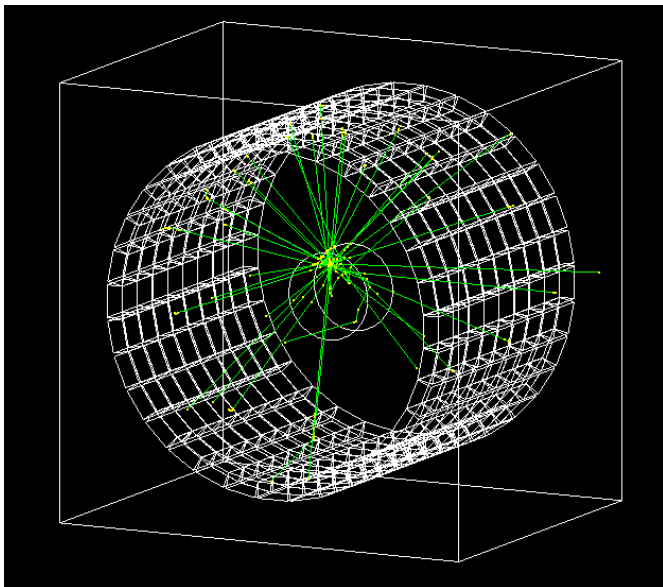
Obtaining the Output ROOT File

GEANT4 based program can save results obtained by the Sensitive Detector in ROOT data file. Example script that takes demonstrates it is `/examples/B3/B3a`.

Running the B3a example

Example schematically simulates a Positron Emitted Tomography system.

- Scintillating crystals are made of Lu_2SiO_5 and arranged to form a ring.
- The head of a patient is modeled as cylinder of brain tissue.



Observable values from sensitive detectors are output in the ``scoring.root`` file. To visually inspect the output data and make simple plots use in ROOT's TBrowser:

```
Root          # enter ROOT session
new Tbrowser() # launch ROOT file browser
```

Note. Utilize full ROOT potential by opening the output ``.root`` file in a custom custom ROOT script. Some [C++ knowledge is required](#).

Summary

- We learned some fundamentals about *secure shell* (SSH) and operating systems' *environment variables*. Obtained skills were applied to login to the Computing Environment and utilize ROOT and Geant4 software.
- Basic Geant4 example “B1” was compiled from source and linked with the frameworks' libraries into an executable file. Gamma rays were injected in detector geometry. Difference between the Geant4 interactive and batch modes is studied.
- To be continued... using ROOT to process GEANT4 simulation data.

New terminal commands utilized in this tutorial

```
ssh -y username@hostname      # establish ssh connection
Env                            # print environment variables
<command> | grep <keyword>    # filter the output of a command by keyword
export VAR=$VAR:<path>        # append to environment variable
source <script>               # “allow” script to change your environment
<command1> && <command2>      # shorthand: execute command2 after command1
cmake                          # invoke Cmake generator to produce Makefile
make                           # compile and link code; produce executable
```

Homework

Geant4 practice

- Check out the [GEANT4 YouTube Mini Course](#) by Jing Liu. Tutorial covers installation of GEANT4 on your local computer and basic framework use.
- Login to the Computing Farm environment.
Copy Geant4 examples to your home folder.
Pick a GEANT4 example from `basic` folder.
Get familiar with the example's README file.
Compile and run example.
Familiarize yourself with Geant4 interactive GUI. Shoot a few events. Rotate and move camera etc...

ROOT practice

Check out this [5-day ROOT tutorial](#). In addition to Nilanga's introduction to the ROOT scripting language this tutorial also covers the Graphical User Interface (GUI) of ROOT. You do not have to go through all five days. Things can get a little challenging along the way.