Introduction to Scientific Computing (MA305)

Dr. Harihar Khanal
harihar.khanal@erau.edu

Department of Mathematics
Embry-Riddle Aeronautical University, FL

Fall 2018

Lec 1. Introduction, Unix Account Setup

What is Scientific Computing?

- **A tool for Computational Sciences**
  Physical problem $\rightarrow$ *Mathematical model* $\rightarrow$ *Algorithm* $\rightarrow$ *Computer code*
- **A cross-disciplinary field intersecting**
  - **Applied Mathematics:** Mathematical modeling of real life phenomena
  - **Numerical Analysis:** Algorithmic thinking of mathematical models
  - **Computing:** Implementation of the numerical algorithms, visualization and presentation of the results

This course primarily focuses on the computing area

Introduction to Unix, Python, NumPy, Matplotlib, SciPy, Latex

Relevant Books

- Learning Scientific Programming with Python by Christian Hill, Cambridge University Press, 2015.
- Elements of Scientific Computing by Tveito, Aslak et al., Springer, 2011.
- Introduction to High Performance Scientific Computing by Victor Eijkhout, 2010.

Useful Web Resources

- UNIX Tutorial for Beginners
- Intro to Linux free edX course by Linux Foundation
- Unix Guru Universe - Beginners & Users
- PUTTY ans PSCP
- OpenSSH
- Think Python
- Latex Help

Syllabus, Tentative Course Calender

Unix is an operating system typically used by academic and scientific researchers. It was originally developed by AT&T during 1969-1977 and further expanded at UC Berkeley. **Linux**, an open-source operating system modeled on Unix, has become popular with a variety of users. Mac also has Unix!

Why Unix/Linux?

- Most of the computing resources for large simulations use Unix-like systems (Linux, Mac OS X Darwin).
- Source code for Unix is publicly available.
- A variety of commercial/public domain software is available.
- Virtually every kind of computer supports Unix.
- Powerful native tools to manage and process complex dataset, ideal for managing systems with large number of users!

www.unix.org, www.linux.org, www.aple.com/macosx/features/unix

About Programming Language

- C/C++, Fortran, Matlab, and Python are widely used in scientific computing.
- C/C++ and Fortran are compiled languages whereas Matlab and Python are interpreted. Fortran and C are recommended for faster computations.
- With the introduction of NumPy and SciPy libraries that use the compiled C code "under the hood", Python has been increasingly popular in scientific computing community.
- C/C++ and Fortran are statically typed languages whereas Matlab and Python are dynamically typed.

Why Python?

- It is free.
- It is relatively simple to learn.
- It has cross-platform support: Windows, Linux and Mac OS X
- It has a large library of modules and packages that extend its functionality.

Unix account setup

- All students in this class should have access to ERAU Unix server (webcat.db.erau.edu), a Linux cluster (wxsession.db.erau.edu) at Weather Lab, and to all machines in this classroom and Computational Math Lab (COAS 301.39). If you do not have your unix account **yet**, use the following link to create your account **now**.

  https://dbwebcat.erau.edu/unix_acct_request.php

  - **Login:** All machines in this classroom (COAS 127) have dual booting system with Microsoft Windows and Redhat Linux. Now restart your computer and login with your Unix account username and password. You have 6 seconds to choose Linux at the boot time. The default setting is for Windows.
  - **Logout:** To shut down your computer click on *System Setting* on the panel on the top of your screen and select shut down in the drop down menu. You have not created any files today, so no need to worry for saving documents!

Introduction to Unix/Linux

. Unix OS is made up of three parts: *kernel*, *shell* and *programs*.

- The core program of Unix is called the kernel. It takes care of all the low level communications between running processes and the hardware, and it must run continuously while the computer is in operation.
- A program is a file containing a collection of instructions to be executed. An instance of running a program is called a process, it is identified by a unique process identifier (PID). Init process PID=1, executed by the kernel, is in charge of controlling and managing other processes.
- Shells are processes that act as the interface between a user and the kernel. It interprets the commands the user types in and arranges for them to be carried out.

Terminal windows and text consoles

- The Unix commands (you type in computer) are executed by an interpreter called *shell*. There are a few Unix shells available sh, bash, csh, tcsh, ksh, z and many commands are common. Most Unix-like systems use *bash shell* by default.
- Were do we type the Unix commands? We need a terminal window and a text console.
- To launch a text console (Xterminal session) click on the xterm icon in the panel or from the drop-down menu: *Application* (top left on the panel) $\rightarrow$ *System* $\rightarrow$ *Terminal*. An xterm window will appear with a Unix prompt ($\sim$ \$[]), waiting for you to start entering commands.
- Alternatively you can dispense with the graphical environment entirely and go back to the text-mode Unix. Press [Ctrl]+[Alt]+[F2] to get a pure text login console. [Ctrl]+[Alt]+[F1] returns you to the graphical interface.

Some basic commands

- To change password type 'passwd' in the Unix prompt and press the Enter key.
  `username:~$ passwd`
- Try some commands:
  `username:~$ date`
  `username:~$ cal`
  `username:~$ who`
  `username:~$ whoami`
  `username:~$ eject`
  `username:~$ exit`
- To log out or quit the Xterminal click in the [X] button at the upper right corner or use the commands: `exit`, or [Ctrl]+[D]
- Next, we will learn Unix file system and some basic Unix commands for listing, moving, creating and editing files.