



Python Computing: Building a Sensor System

CSCI 250

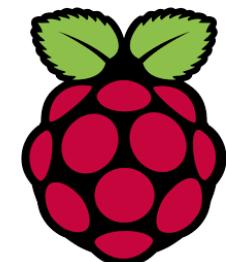
Lecture 3: Linux Day – Basic Commands



COLORADO SCHOOL OF MINES
EARTH • ENERGY • ENVIRONMENT

Today's Plan

- + Quick Review
- + Operating System
- + Linux Commands
- + Submissions
 - + Setup Rpi – Done!
 - + Linux Practice
- + Anyone still need kit?



Learning Outcomes

- + By the end of this course, students will be able to:
 - + **Create, navigate, and manage files and directory structures using basic Linux shell commands.**
 - + Describe the functionality and purpose of the individual components of the Raspberry Pi Hardware.
 - + Install the Raspbian operating system onto the Raspberry Pi Hardware and setup basic configuration parameters.
 - + Download, install, and develop programs using an Integrated Development Environment (IDE) on the Raspberry Pi Hardware.
 - + Develop and run basic Python functions and programs in the Linux environment to collect data from sensors using the Raspberry Pi Hardware (e.g., optical, acoustic, acceleration, magnetic field).
 - + Plot and analyze data from the sensor system and compare to mathematical models.

Programming Environment

- + Integrated Development Environment (IDE)
 - + Spyder IDE (Book Appendix A)
 - + IDLE - Preloaded on the Rpi
 - + Easy to use
- + IDE is good to start with ... will use IDLE in class
 - + Recommended not required
- + Can do everything in terminal mode
 - + Text editor (e.g., pico, nano, vi)
 - + Interpreter (shell commands)



Getting started with Python

- + Before getting started with the sensors
- + IDLE
 - + Open IDLE
 - + Interactive
 - + Batch
- + Chapter 1 –
 - + Algorithmic thinking – breaking down problems into sub problems
 - + Sequence of instructions or steps
 - + Skip the Spyder specific information – replace with Python 3 (IDLE)
 - + Very basics, importing libraries, variable names, functions





Python

- + Most popular and best supported for Rpi
- + Version 2 vs. 3 (Book Appendix C)
- + Interpreted language
- + Advantages
 - High level language, easy to learn – details abstracted
 - Don't have to explicitly declare data types
 - No memory management needed – no pointers
 - Object oriented programming, uses classes
- + Disadvantage: Slow

Chapter 1 Review

- + Algorithmic thinking
 - + Breaking down problems into sub problems
 - + Sequence of instructions or steps
- + Skipped the Spyder specific info - replace with Python 3 (IDLE)
- + Very basics
 - + Importing libraries
 - + Variable names
 - + Functions
 - + Printing



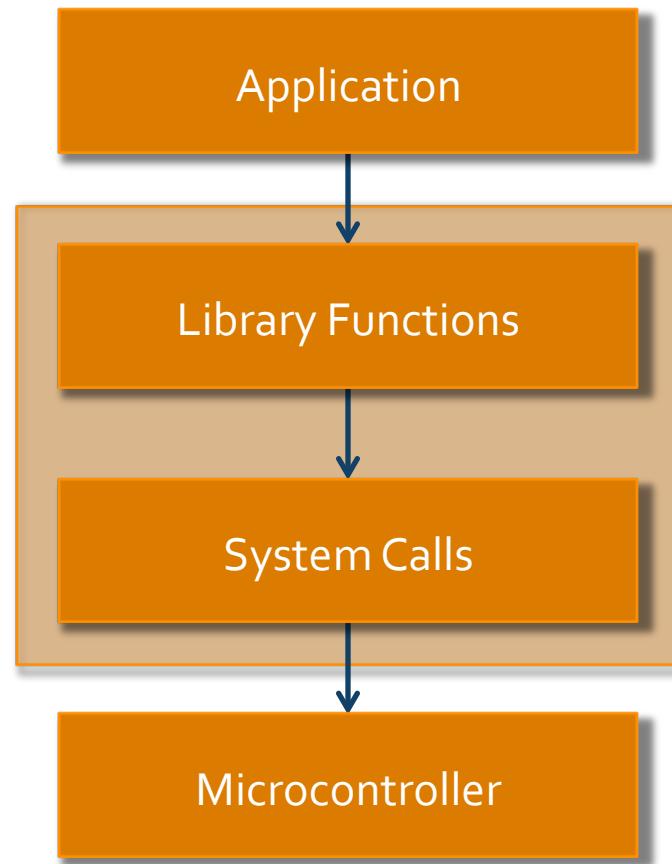
```
#This is a comment, import a library
import numpy as np

#Create a variable, assign a value
myVariable = np.sqrt(25)

#print the value and a message
print(myVariable)
print("Python ROCKS!")
```

Operating System

- + Operating system – simplified
- + Advantages
 - + User interface (text or graphic)
 - + File system
 - + Multiple processes
 - + Use of hardware devices easily
- + Disadvantages
 - + More overhead, uses more power
- + Raspbian OS – Linux based



Linux Basics

- + sudo shutdown
- + Help
 - + man
- + File/Directory Basics
 - + ls, cp, mv, cd, pwd, mkdir
- + File Viewing/Manipulation
 - + cat, head, tail, grep
- + Advanced Topics
 - + File permissions, process control, networking, etc.



Practice Time

- + Linux practice example ... you can customize :~)
 - + Open terminal or be in terminal mode
 - + Use the man command to get help on a command (any)
 - + Enter the command to show the current directory (pwd)
 - + List the contents of the current directory (ls and ls -l)
 - + Create a new directory (mkdir)
 - + Navigate into that directory (cd)
- + Submit photo on Canvas
- + When complete ... check advanced commands and Linux Quick Start Guide on Canvas or start pre-reading Ch. 2.1-2.4

Wrap Up

+ Discussion:

- + Quick review, high level operating system discussion, and basic Linux commands.

+ Assignment:

- + Linux Practice – Photo (due tonight by 11:59)
- + Pre-read Chapter 2.1-2.4

+ Next week

- + Sensor Day
 - + Bring your “What’s in the Box” worksheet
- + Begin First Lab and Chapter 2

