Introduction to **Linux**

Augustine Abaris
Research Computing Services
Information Services & Technology

BOSTON UNIVERSITY

Topics for Today

- Research Computing Services
- Linux Overview
- Linux Interaction - Shell and Commands
- I/O redirection (pipes, etc.)
- Navigating the file system
- Processes and job control
- Editors
- Creating and Running Code

Research Computing Services

Research Computing Services (RCS)

A group within Information Services & Technology at Boston University provides computing, storage, and visualization resources and services to support research that has specialized or highly intensive computation, storage, bandwidth, or graphics requirements.

Three Primary Services:

- Research Computation
- Research Visualization
- Research Consulting and Training

RCS Team and Expertise

Our Team

- Scientific Programmers
- Systems Administrators
- Service Management Specialists
- Research Facilitators
- Special Initiatives (Grants)
- help@scc.bu.edu

Consulting Focus:

- Bioinformatics
- Data Analysis / Statistics
- Molecular modeling
- Geographic Information Systems
- Scientific/Engineering Simulation
- Visualization

Me

- Lead Systems Programmer/Administrator
- 20+ years of systems administration experience
- 15+ years at BU, contributed to design and deployment of SCC and 2 predecessor HPC clusters
- Contact: augustin@bu.edu

You

- Who has experience programming?
- Using Linux?
- Using compute clusters?
- Using the Shared Computing Cluster (SCC)?

Linux

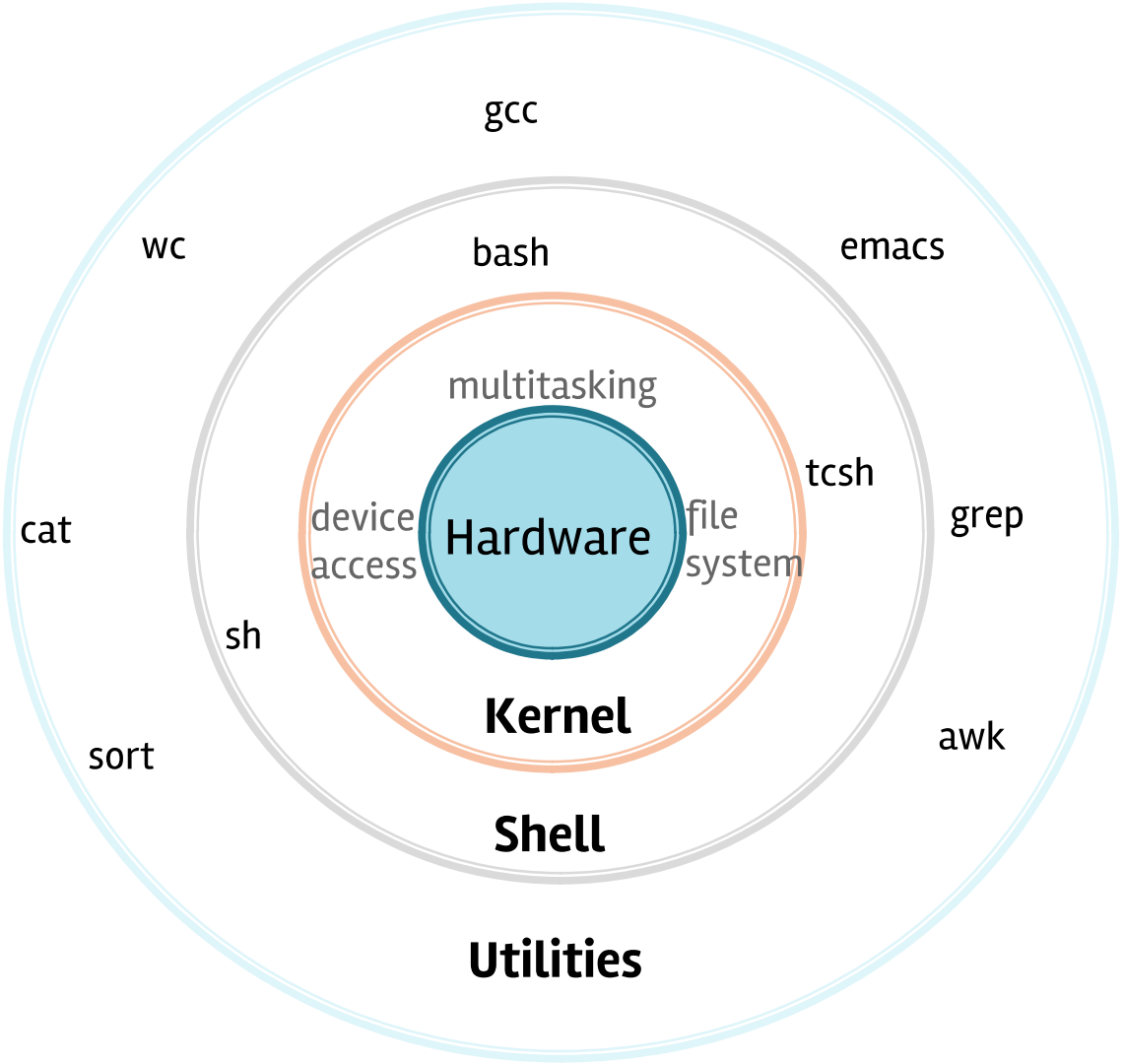
What, Who, When, Where & Why

What is Linux

- Unix-like computer **operating system** assembled under the model of free and open-source software development and distribution.
- These operating systems share the **Linux kernel**.
  - Typically have the GNU utilities
- Comes in several “distributions” to serve different purposes.

What is Linux

- Bird's eye view



Who is Linux

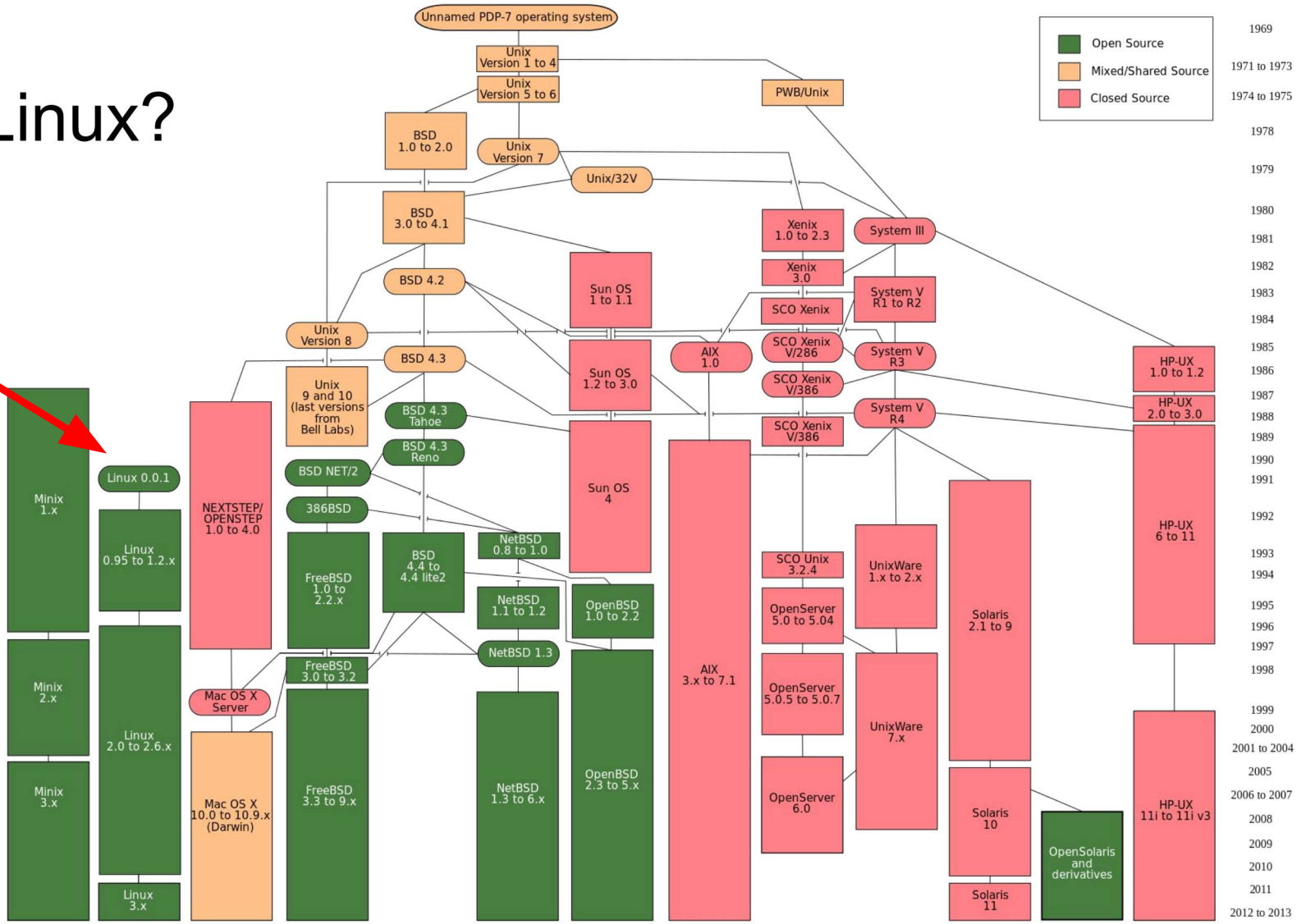
- Linux is an O/S core originally written by Linus Torvalds. Now almost 10,000 developers including major technology companies like Intel and IBM.

&

- A set of programs written by Richard Stallman and others. They are the GNU utilities.

When is Linux?

~1991



Where is Linux

- World Wide Web
  - 67% of the world's web-servers run Linux (2016)
- Research/High-Performance Compute
  - Google, Amazon, NSA, 100% of TOP500 Super-computers.
- Modern Smartphones and devices
  - The Android phone
  - Amazon Kindle
  - Smart TVs/Devices

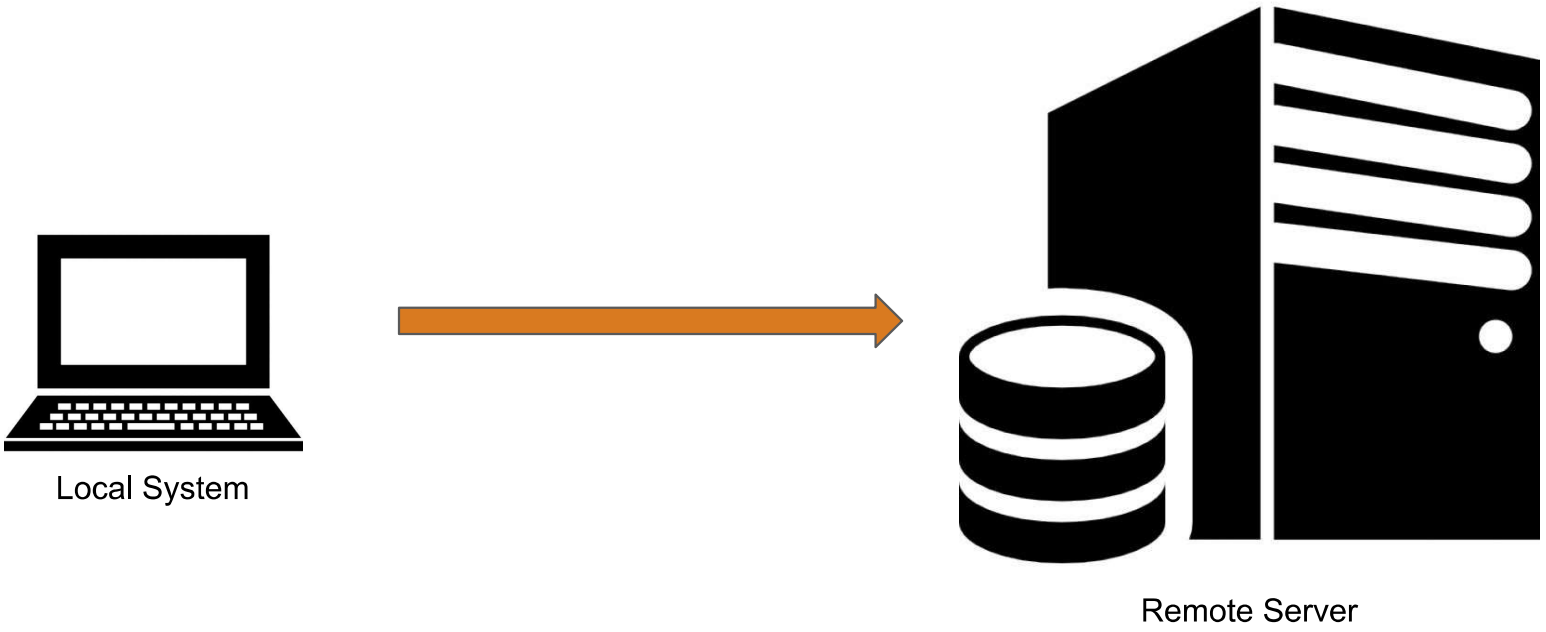
Why Linux

- Free and open-source.
- Powerful for research datacenters
- Personal for desktops and phones
- Universal
- Community (and business) driven.

The most common OS used by BU researchers when working on a server or computer cluster

Connecting

Let’s use Linux

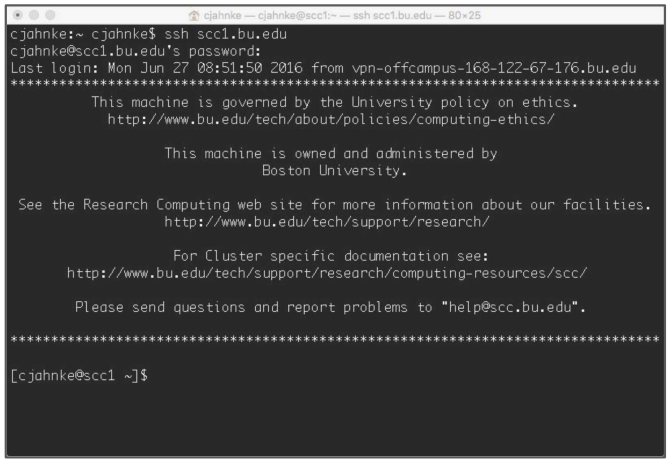

Local System
Remote Server

Connection Protocols and Software

Remote Connections:

Secure **SH**ell

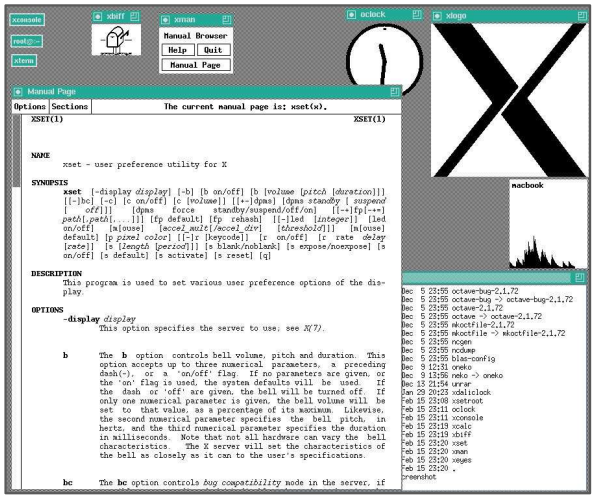
(SSH)



Remote Graphics:

X-Windowing

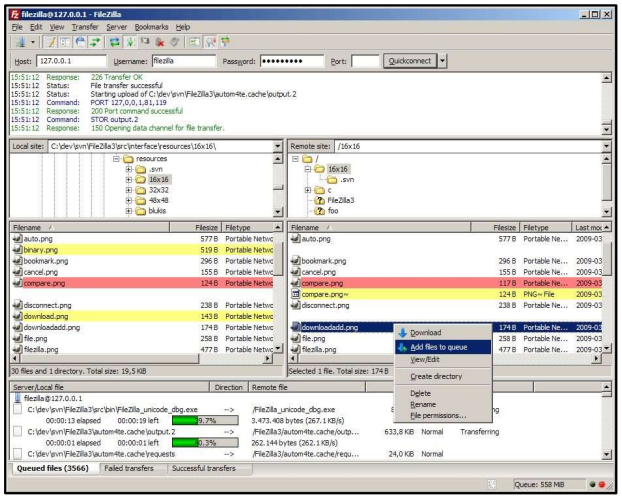
(X, X-Win)



Data Transfer:

Secure **F**ile **T**ransfer **P**rotocol

(SFTP)



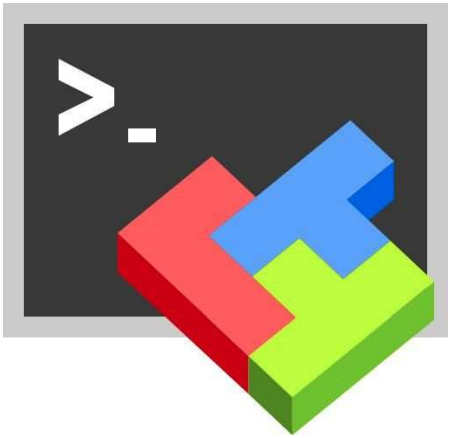
Other protocols too, but let's start with these.

Connecting from Different Platforms

	SSH	X-Win	SFTP
Microsoft Windows		**MobaXterm** https://mobaxterm.mobatek.net	
Apple macOS	**Terminal** (Built in)	**XQuartz** https://www.xquartz.org	**Cyberduck** https://cyberduck.io
Linux	**Terminal** (Built in)	**X11** (Built in)	**Various** (Built in)

SCC Help: http://www.bu.edu/tech/support/research/system-usage/getting-started

Microsoft Windows



You need software that emulates an “X” terminal and that connects using the “SSH” Secure Shell protocol.

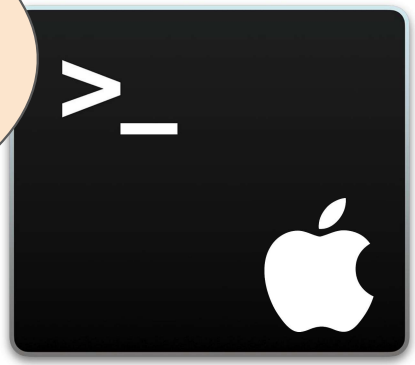
- **Recommended**: MobaXterm
  - Download: http://mobaxterm.mobatek.net/

- Alternatives:
  - SSH/X-Windows: X-Win32
    https://www.bu.edu/tech/services/support/desktop/distribution/xwindows/
  - SFTP: Filezilla
    https://filezilla-project.org/

SCC Help: http://www.bu.edu/tech/support/research/system-usage/getting-started/connect-ssh

Apple macOS

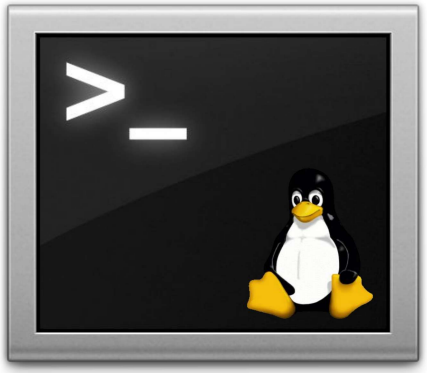
Built in!
Apple macOS is built on Darwin -- a derivative of 4.4BSD-Lite2 and FreeBSD



- SSH: Terminal
  - Built in to macOS
    Applications > Utilities > Terminal
- X-Windows: XQuartz
  - Download: https://www.xquartz.org/
  - Note: This install requires a logout.
- SFTP: Your choice
  - Filezilla: https://filezilla-project.org/ (Cross-platform, open-source)
  - Cyberduck: https://cyberduck.io (macOS native, drag-and-drop)
  - Many others

SCC Help: http://www.bu.edu/tech/support/research/system-usage/getting-started/connect-ssh

Linux



- SSH: Terminal
  - Built in to Linux
    Applications > System > Terminal
- X-Windows: X11
  - Built in to Linux
  - Use your package manager.
- SFTP: Your choice
  - Usually has one Built in.
  - Alternate: Filezilla (https://filezilla-project.org/)

SCC Help: http://www.bu.edu/tech/support/research/system-usage/getting-started/connect-ssh

Connecting

- Use your Shared Computing Cluster account if you have one.

- Tutorial accounts if you need one.
  - Username:
  - Password:

Tutorial credentials blocked for print.
This box disappears during presentation

```
[local_prompt]$ ssh username@scc1.bu.edu
```

Get supplementary files

- At the command prompt, type the following:

```
[username@scc1 ~]$ cd ~
[username@scc1 ~]$ tar xf /scratch/linux-materials.tar
[username@scc1 ~]$ ls
c   data   haystack   scripts
```

Linux Interaction

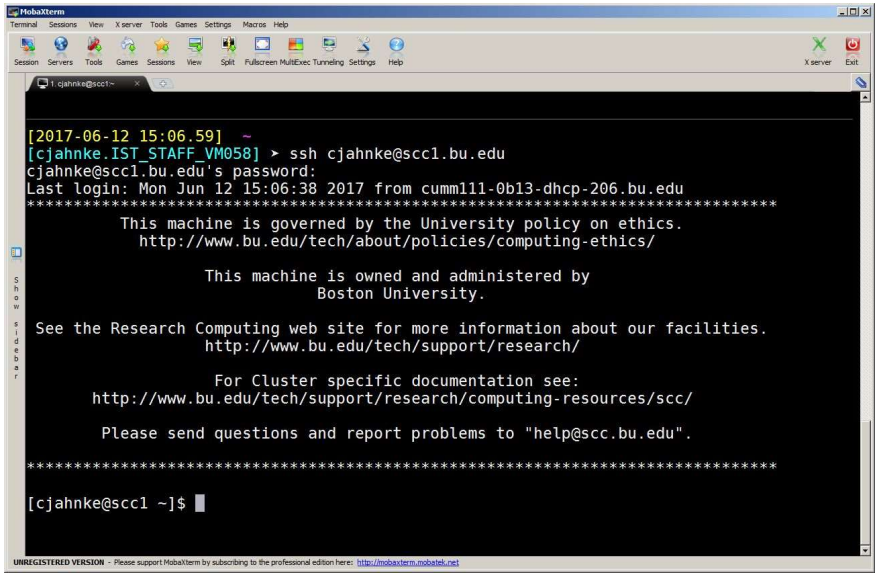
Shell, Prompt, Commands and System Use

Linux: The Shell

- Program that ***interprets commands*** and sends them to the OS

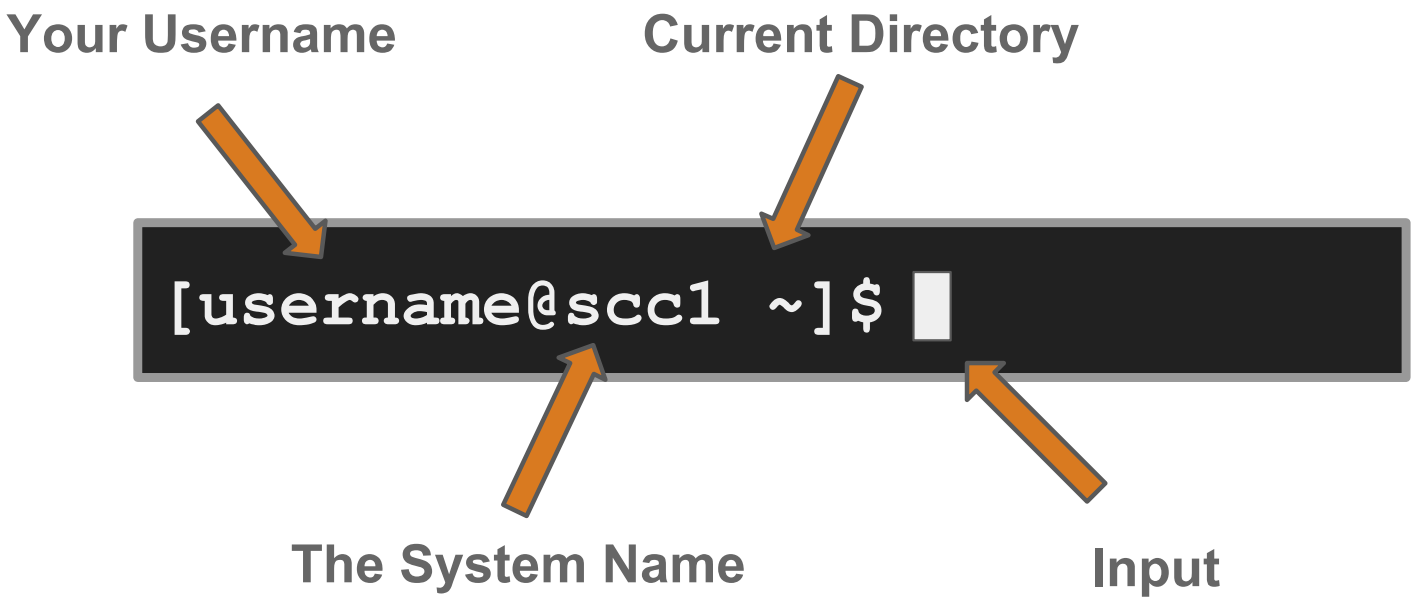
- Provides:
  - Built-in commands
  - Programming control structures
  - Environment variables

- Linux supports multiple shells.
  - The default on SCC is **`Bash`**.



"**`Bash`**" = "**B**ourne-**a**gain **Sh**ell"
(GNU version of ~1977 shell written by Stephen Bourne)

Linux: The “prompt”



(In Linux “ ~ ” is a shorthand for your home directory.)

Linux: Command Basics

```
[username@scc1 ~]$ command --option argument
```

- **Command**: Command/program that does one thing
- **Options**: Change the way a command does that one thing
  - Short form: Single-dash and one letter e.g. **`ls -a`**
  - Long form: Double-dash and a word e.g. **`ls --all`**
- **Argument**: Provides the input/output that the command interacts with.

For more information about any command, use **`man`** or **`info`** (e.g. "**`man ls`**")

Commands: Hands-On

- After you connect, type
  - `whoami` # my login
  - `hostname` # name of this computer
  - `echo “Hello, world”` # print characters to screen
  - `echo $HOME` # print environment variable
  - `echo my login is $(whoami)` # replace $(xx) with program output
  - `date` # print current time/date
  - `cal` # print this month’s calendar
  - `shazam` # bad command

Commands: Hands-On Options

- Commands have three parts; command, options and arguments/parameters.

  Example: cal –j 3 1999. “cal” is the command, “-j” is an option (or switch), “3” and “1999” are arguments/parameters.

```
[username@scc1 ~]$ cal -j 3 1999
```

- What is the nature of the prompt?
- What was the system’s response to the command?

Commands

"Small programs that do one thing well"

- The Unix Programming Environment, Kernighan and Pike

  … at its heart is the idea that the power of a system comes more from the relationships among programs than from the programs themselves. Many UNIX programs do quite trivial things in isolation, but, combined with other programs, become general and useful tools.