Com S // Cpr E // Math 5250

Numerical Analysis of High-Performance Computing

Instructor: Songting Luo

Lecture 2: Unix and a Python Example

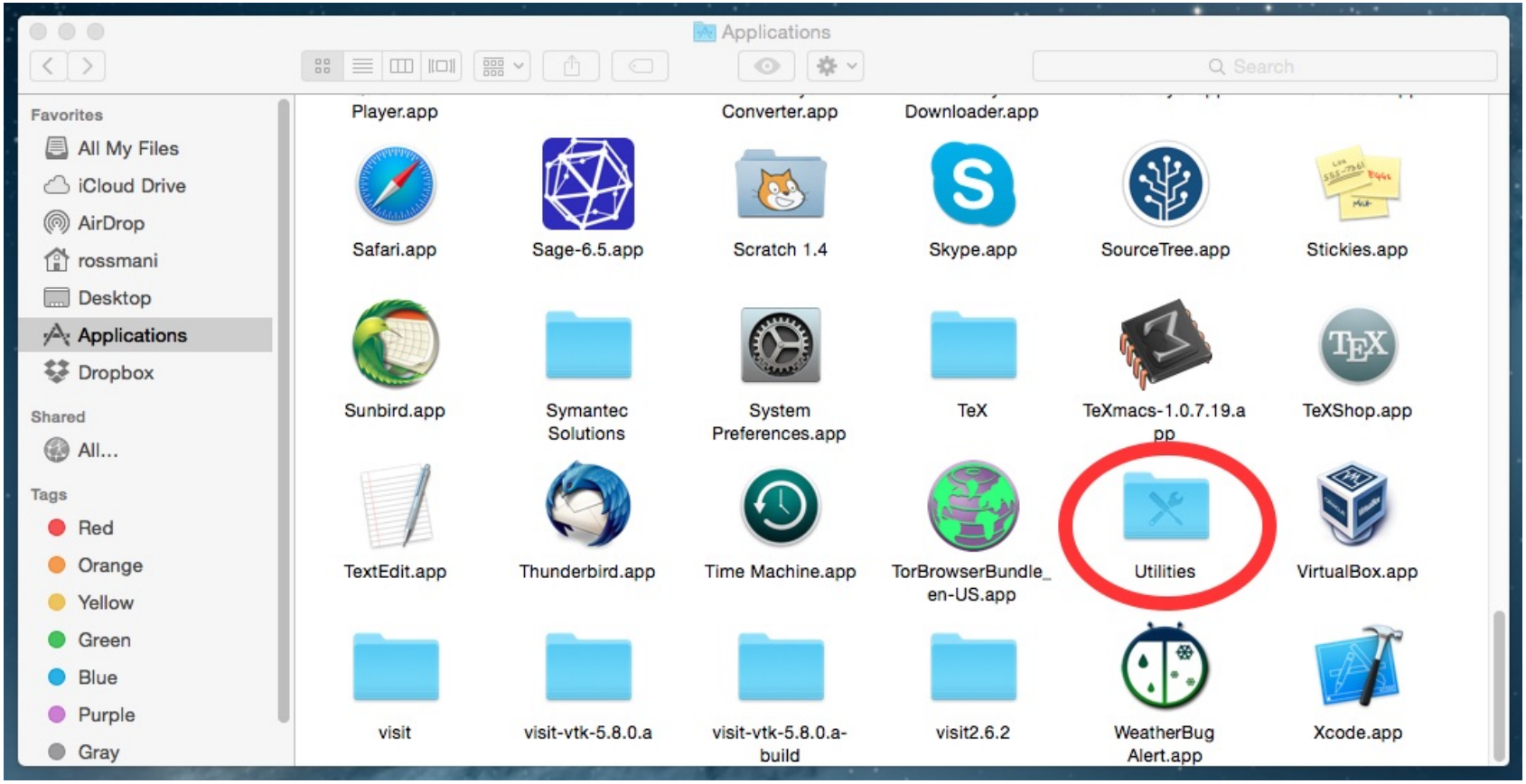
Outline

1. The Terminal Window and Unix
2. Git and GitHub
3. Python example

The Terminal Window and Unix

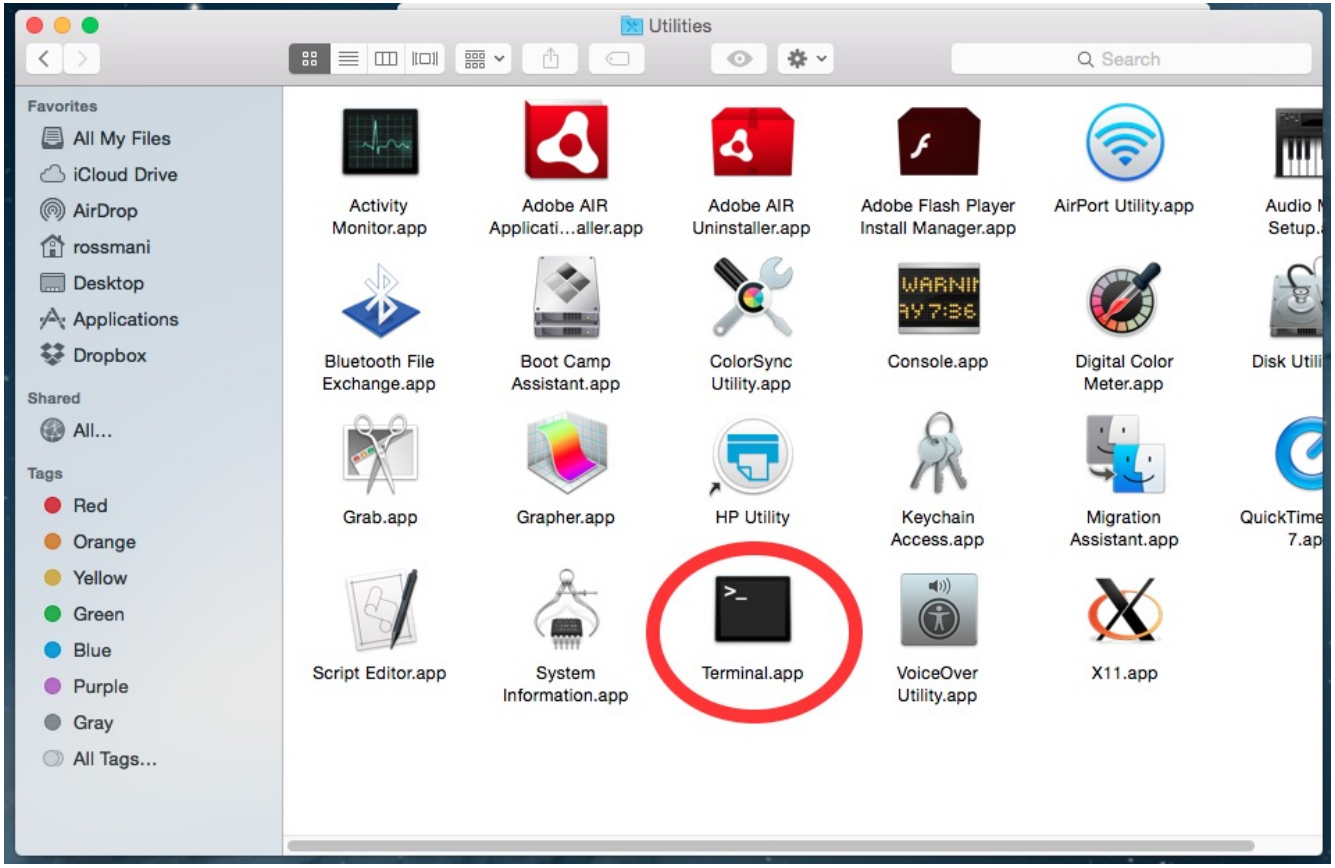
Terminal

The Terminal application lets us directly use command line Unix



Terminal

/Applications/Utilities/Terminal.app



Basic unix commands

- `pwd` – find out what directory you are in
- `ls` – see what files you have in the current directory
- `cd some/directory` – change directory
- `cp somefile newfile` – for copying one file to another
- `mv oldfile newfile` – for moving and renaming files
- `rm somefile` – remove a file
- `rm -r somedirectory` – remove recursively (used to delete directories)
- `mkdir newdirname` – create directory
- `more somefile` – use to scan through ascii files
- `vim somefile` – vi text editor to edit ascii files
- `emacs somefile` – emacs text editor to edit ascii files

DANGER: There is no simple undo in Unix. If you remove a file via `rm`, it will be gone forever. A better idea is to use `rm -i`, in which case you get a question asking if you really want to delete this file.

NOTE: Use tab-complete to reduce the amount of typing. Type only first part of a file or directory name, then press TAB to auto-complete the full name.

Basic unix commands

Quick Unix Demo ...

Introduction to Unix and Tutorials

Introduction to Unix and Tutorials

Git and GitHub

Version control systems

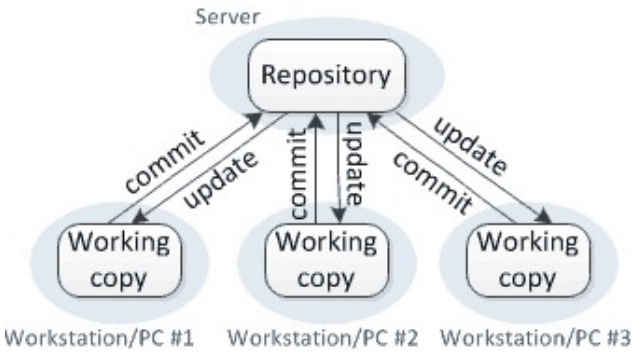
Originally developed for large software projects with many developers

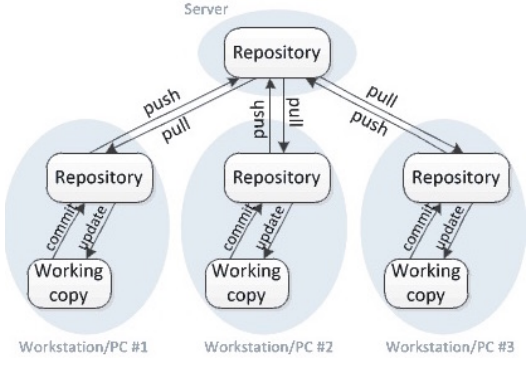
Also useful for single user:

- Keep track of history and changes to files
- Be able to revert to previous versions
- Keep many different versions of code well organized
- Easily archive exactly the version used for results in publications
- Keep work in sync on multiple computers

Two types of version control models

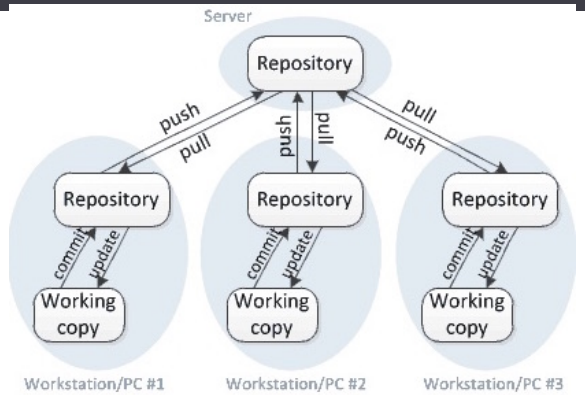
- Server-client model: CVS, Subversion (aka SVN)
- Distributed version control: Git, Mercurial





https://homes.cs.washington.edu/~mernst/advice/version-control.html

Distributed version control: Git, Mercurial

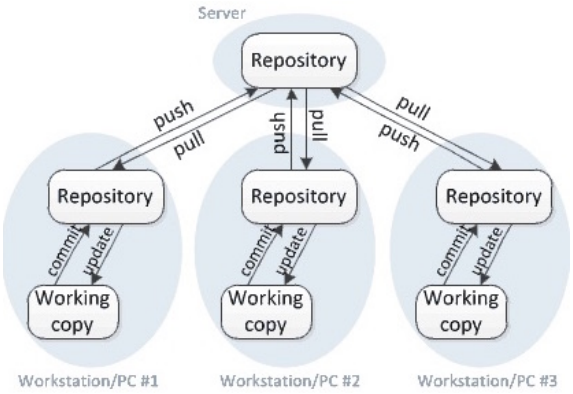


https://homes.cs.washington.edu/~mernst/advice/version-control.html

- When you git clone a repository you get all the history (stored in .git)
- `$ git clone {repository on GitHub} ISU-HPC`

  will make a complete copy of the class repository and call it **ISU-HPC**
- Git commands:
  - git commit – commits to your clone's .git directory
  - git push – sends your recent changesets to another clone (by default 'another clone' = the repository you cloned from GitHub), but you can push to any other clone (with correct permissions)
  - git fetch or git pull – pulls changesets from another clone
  - git merge – applies changesets to your working copy
- GitHub, Bitbucket, and GitLab are popular cloud-based code repositories.

Distributed version control: Git, Mercurial



https://homes.cs.washington.edu/~mernst/advice/version-control.html

Advantages of the distributed model:

- You can commit changes, revert to earlier versions, examine history, ... without being connected to server
- Also without affecting anyone else's version if working collaboratively – can commit often while debugging
- No problem if server dies, every clone has full history

NOTE: for collaboration will still need to push or fetch changes eventually and git merge may become more complicated

Python example

Computing the square root

- Hardware arithmetic units can add, subtract, multiply, and divide
- Other mathematical functions require software

Compute $\sqrt{2} \approx 1.4142135623730951$

In most languages, there exists a function called sqrt that computes this. In Python:

```
>>> from numpy import sqrt
>>> sqrt(2.0)
1.4142135623730951
```

One possible algorithm to approximate $s = \sqrt{x}$

A simple iterative scheme

```
s = 1.0       # or some better initial guess
for k in range(kmax):
    s = 0.5 * (s + x/s)
```

where `kmax` is some maximum number of iterations

- In Python, range(N) is $[0, 1, 2, \ldots, N-1]$
- Why does this work?

$$\text{If} \quad s < \sqrt{x} \quad \text{then} \quad x/s > \sqrt{x}$$
$$\text{If} \quad s > \sqrt{x} \quad \text{then} \quad x/s < \sqrt{x}$$

- In fact this is Newton's method to find root of $s^2 - x$

Newton's method

Problem: Find a solution of $f(s) = 0$ (zero or root of f)

Idea

Given an approximation (i.e., a good guess to the root): $s^{(k)}$
approximate $f(s)$ by a linear function:

$$f(s) = 0 \approx f\left(s^{(k)}\right) + \left(s - s^{(k)}\right) f'\left(s^{(k)}\right) \quad \Longrightarrow \quad s \approx s^{(k)} - \frac{f\left(s^{(k)}\right)}{f'\left(s^{(k)}\right)}$$

Newton's method

$$s^{(k+1)} = s^{(k)} - \frac{f\left(s^{(k)}\right)}{f'\left(s^{(k)}\right)}$$

Python example: square root function

Goals:

- Develop our own version of `sqrt` function using Newton's method
- Start simple and add complexity in stages
- Illustrate some Python programming

Note: We will do this in

`$ISUHPC/lectures/lecture2`

directory so you can examine the various versions later

Python example: factorial

- We want to calculate $n!$.
- Can be done in a simple for loop.

$n!$ in a simple for loop

```
s = 1
for k in range(1,n):
    s = s*(k+1)
```

Python Lab Assignment: exponential function

- We want to calculate e^x.
- We will store the value of $e \approx 2.7182818284590451$.
- From x we find the nearest integer, let's call it x_0:

```
x0 = int(round(x))
```

- We then compute the Taylor series of e^x about $x = x_0$.

Taylor series for e^x about $x = x_0$

$$z = x - x_0$$

$$e^x = e^{x_0}\left(1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \dots\right)$$

Python Lab Assignment: natural logarithm function

- We want to calculate $\ln(x)$.

$$s = \ln(x) \quad \Longrightarrow \quad e^s = x \quad \Longrightarrow \quad f(s) = e^s - x$$

- Use Newton's method with initial guess $s = x$

Newton's method for $f(s) = e^s - x$ (x is given)

$$\begin{aligned} s^{(k+1)} &= s^{(k)} - \frac{f\left(s^{(k)}\right)}{f'\left(s^{(k)}\right)} \\ &= s^{(k)} - \frac{\exp\left(s^{(k)}\right) - x}{\exp\left(s^{(k)}\right)} \\ &= s^{(k)} - 1 + x\exp\left(-s^{(k)}\right). \end{aligned}$$

Python Lab Assignment

- Develop a Python script with name **demo_myfuncs.py** to test the functions shown above, run it on terminal (Submit both code and screen shots)

  (Text Editors (pick one): Leafpad, gedit, NEdit, XCode, TextWrangler, BBEdit, vi or vim, nano, emacs, ...)
- Set up Git repository ((pick one) GitHub, Gitlab, BitBucket, ...), submit screenshot.