CS 35L Software Construction Laboratory (Lab4-A)
Mon, Jan 30, 2012, Ver 1.1

In-class presentation
  Starting from next Monday (1/30/12), until the last day of class 3/14/12)
  Up to 3 students are allowed to present each day
  5 ~ 10 min for each talk
  The presentation is weighted 5% in the final grade

Linux/Unix login process
  1. Hardware power on, boot load program
  2. Loading system kernel
  3. Starting init process (PID 1)
  4. Wating for user name
      > login:
  5. User providing their password
      > password:
     Note: step 4 and 5 depends on the file "/etc/passwd" which looks like this:
     irc:x:39:39:ircd:/var/run/ircd:/bin/sh
                       (home dir)   (shell app)
  6. Start the shell app

  NOTE: check your default shell app (echo $SHELL)
  More reading: http://www.ucblueash.edu/thomas/Intro_Unix_Text/Process.html

Command "diff" revisited
  diff file_1 file_2
    Note: without any option, diff will give you normal outut
          a stands for added, d for deleted and c for changed

  diff -u file_1 file_2
    Note: with -u option, you will get unified format output
          Programs using this format: patch, git, svn

  Unified format output
      a diff file starts with two lines describe two files
        --- test   2011-10-17 08:51:11.200077000 -0700
        +++ test2   2011-10-17 08:51:30.086132000 -0700
      a diff file contains one or more change hunks starting with
        @@ -l,s +l,s @@
      The first pair of l,s is for the origin file, and the second pair for the
    new file. l indicates the start line number while s is the number of lines
    in the hunk.

  More reading: http://en.wikipedia.org/wiki/Diff

Command "patch"
  apply a diff file to an original
  usage:
     1/ patch [options] [originalfile [patchfile]]
     2/ patch -p0 <patchfile
  Note: Read Manpage to understand what's the difference between -p0 and -p1

Lab 4 Guideline

NOTE: This is a new lab which was just updated yesterday
In this lab, you need to create a patch using git and then apply it to a previous version of the source code of diffutil

Software Version Control
Common features of Revision Control
1/ roll-back feature. (a snapshot of the source code)
2/ synchronization between team members
3/ source code distribution (e.g. github)
4/ quality assurance (QA) and code review

More reading: http://en.wikipedia.org/wiki/Revision_control

Two tools we will cover in Thursday: git and subversion (a.k.a. svn)
git http://en.wikipedia.org/wiki/Git_(software)
svn http://en.wikipedia.org/wiki/Apache_Subversion