

CS35L Software Construction Laboratory

Lab 5: Sneha Shankar
Week 4; Lecture 1

Assignment 10 SignUp!

- Signup Link:
https://docs.google.com/spreadsheets/d/1RySZTzefFPbxp48mbW6UJuqsC9ZxEI_TTg8UkD1aPzE/edit?usp=sharing
- Presentations starting next week
 - Time and topic allocation
 - Presentations on Tuesdays/Thursdays
 - Specify the chosen topic in the available time slot in the link above (Topic will be FCFS)
- ~10 minutes presentation
- Details: <http://web.cs.ucla.edu/classes/winter18/cs35L/assign/assign10.html>

Assignment 7 Reminder

Get your BeagleBone

If you are planning to get it from someone, make sure you reset it.

You will be asked to submit the stepwise guide to how you did the reset.

Extra efforts required!

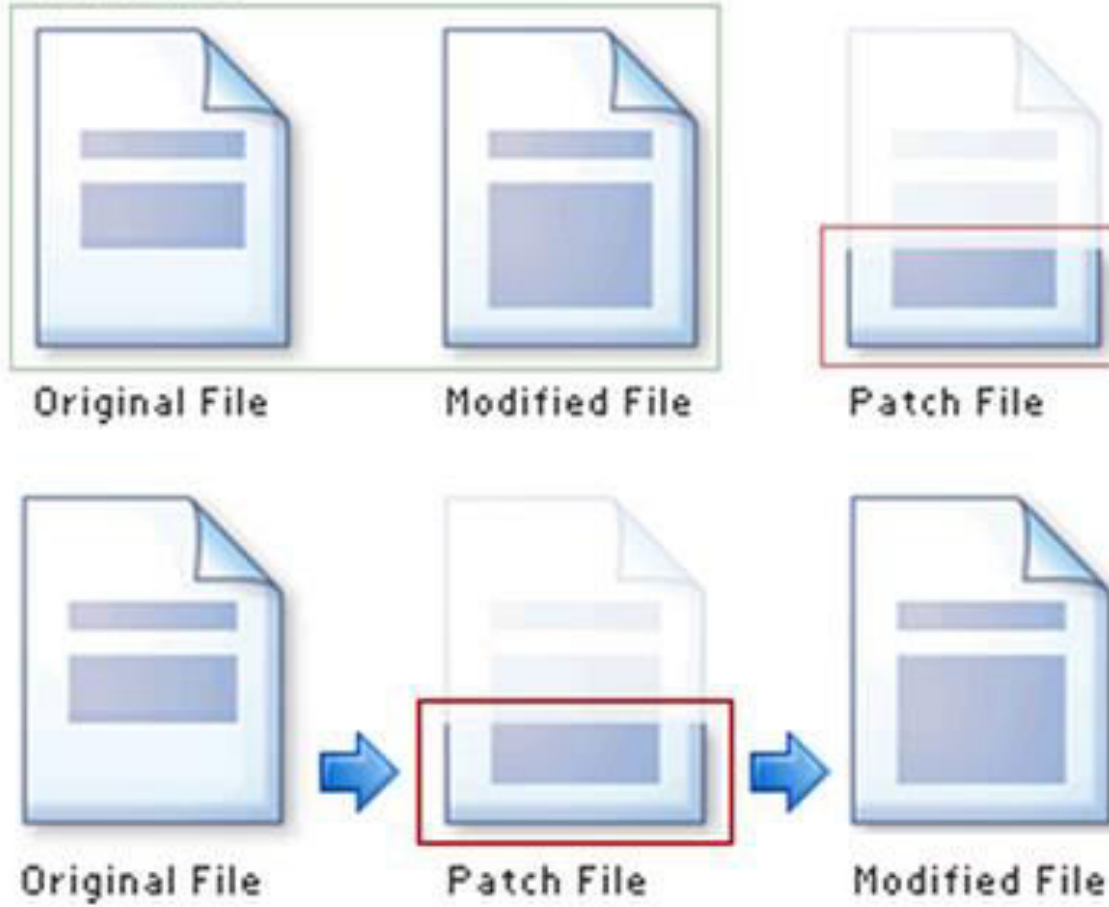
Recommended: Buy a new one

Patching

- A **patch** is a piece of software designed to update a computer program or its supporting data, to fix or improve it.
- E.g. Fixing security vulnerabilities and other bugs with such **patches** usually called bug fixes, and improving the usability or performance.
- It is a .diff file (or patch file) that includes the changes made to a file
- A person who has original (buggy) file can use patch command with the diff file to add the changes to the original file
- Patch command:
 - Usage: patch [options] [original file] [patch file]
 - -pnum: strip the smallest prefix containing num leading slashes from each file name found in the patch file

Applying a Patch

Source Files



Diff format

- `diff -u original_file modified_file`
- `--- path/to/original_file`
- `+++ path/to/modified_file`
- `@@ -l,s +l,s @@`
 - `@@`: beginning of a hunk
 - `l`: beginning line number
 - `s`: number of lines the change hunk applies to for each file
 - A line with a:
 - `-` sign was deleted from the original
 - `+` sign was added to the original
 - stayed the same

Task 1

- Create a file file1.txt with the contents Hello World
- Copy contents of file1.txt to file2.txt
- Add one more line to file2.txt
- Apply diff -u to file1.txt and file2.txt and save the output in a file called patch_file.txt
- Run the patch command on it (patch < patch_file.txt)
- Open file1.txt. What do you see?

Task 2

- Tar file1.txt file2.txt and patch_file.txt together in a tar ball called patched.tar.gz
- Delete the above three files from your directory
- Untar patched.tar.gz

Fixing a bug!

- On a certain computer (not necessarily seasnet), the command `ls -l /bin/bash` displays:
\$ `ls -l /bin/bash`
`-rwxr-xr-x 1 root root 729040 2009-03-02 06:22 /bin/bash`
- But this is a bug! The user wants it to display:
\$ `ls -l /bin/bash`
`-rwxr-xr-x 1 root root 729040 Mar 2 2009 /bin/bash`
 - Traditional linux format
- The problem is with `ls` output, specifically the date portion

Ls output

- Let's investigate ls
 - man ls
- ls -l --time-style=long-iso /bin/bash
 - Outputs the 'buggy result'
- How to reproduce the 'correct' result?

Getting started

- Login to seasnet
- Download coreutils to a temporary directory
 - how can we download a file?
 - 'wget'
- Untar\Unzip it
 - How do you unzip a file?
 - man tar
 - tar -xzvf coreutils-7.6.tar.gz
 - cd into the newly created coreutils folder
 - Make a directory ~/coreutils_install in your home directory (this is where you'll be installing coreutils)
 - mkdir coreutils_install

Tar commands

- `tar -cvf <tarfilename.tar> <target directories>` - creates tar file.
- `tar -tvf <tarfilename.tar>` - list tar file contents
- `tar -xvf <tarfilename.tar>` - extracts tar file
- Can add `-z` flag for newer LINUX distros with gunzip for automatic compress/decompress (.gz suffix).
- Otherwise try compress command (.z suffix)
- USAGE:
 - Always create tar file in target directory (relative file/directory names)
 - Always list tar file before extracting (insure relative file names)
 - Always extract tar file in target directory (relative file/directory names)
- Example:

```
tar -xzvf ~/bb-1_3a_tar.gz
```

Compiling from scratch

- Common scenario
 - You download a utility from the internet to your unix machine
 - There are no binaries, but source code and makefile is available
 - Compile and build to install it
 - Reading text files in the program folder gives clues how to install the program
 - Usually INSTALL, README, readme.txt, install.txt and so on

Compiling from scratch

- Common scenario
 - The order of compilation is usually:
 - **./configure**
 - **make**
 - **make install**
 - Difference? 'Configure' checks for dependencies, 'make' creates executables (compile), and 'make install' runs the install "target" (we discussed this)
 - man make for more details
 - view Makefile in the programs folder for details
 - e.g. search for "install:"

Configure script

- Designed to aid in developing a [program](#) to be run on a wide number of different computers
- **configure** is application specific
 - software provides it own configure script
- Creates the Makefile
 - Can change default behavior with options
 - **./configure --help** for more info

Compiling coreutils

- Reproduce the bug by running the version of 'ls' in coreutils 7.6
- If you just type ls at CLI it won't run 'ls' in coreutils 7.6
 - Why? Shell looks for /bin/ls
 - To use coreutils 7.6:
 - `cd coreutils_install`
 - `./bin/ls -l`
 - This manually runs the executable in this directory

Bug appears with newly built coreutils

```
Inxsrv07 ~/cs35L/lab3/coreutils/bin]$ ls -l /bin/bash  
-rwxr-xr-x 1 root root 960376 Jul 8 2015 /bin/bash
```

```
Inxsrv07 ~/cs35L/lab3/coreutils/bin]$ ./ls -l /bin/bash  
-rwxr-xr-x 1 root root 960376 2015-07-08 04:11 /bin/bash
```

Applying patch

```
diff --git a/src/lsc b/src/lsc
```

```
index 1bb6873..4531b94 100644
```

```
--- a/src/lsc
```

```
+++ b/src/lsc
```

```
@@ -2014,7 +2014,6 @@ decode_switches (int argc, char **argv)
    break;
```

```
    case long_iso_time_style:
-   case long_iso_time_style:
        long_time_format[0] = long_time_format[1] = "%Y-%m-%d %H:%M";
        break;
```

```
@@ -2030,13 +2029,8 @@ decode_switches (int argc, char **argv)
    formats. If not, fall back on long-iso format. */
```

```
    int i;
    for (i = 0; i < 2; i++)
    {
-       char const *locale_format =
-       dcgettext (NULL, long_time_format[i], LC_TIME);
-       if (locale_format == long_time_format[i])
-       goto case_long_iso_time_style;
-       long_time_format[i] = locale_format;
-   }
+       long_time_format[i] =
+       dcgettext (NULL, long_time_format[i], LC_TIME);
    }
```

```
/* Note we leave %5b etc. alone so user widths/flags are honored. */
```

Applying the patch

Read the patch bug report

lists.gnu.org/archive/html/bug-coreutils/2009-09/msg00410.html

Understand what part of the code is being fixed

Apply the patch

- Just use an editor (eg. emacs, vim).
- `cd coreutils-7.6`
- emacs patch_file: copy and paste the patch content
- `patch -pnum < patch_file`
- 'man patch' to find out what pnum does and how to use it
- type `make` to rebuild patched ls.c
- A new executable ls file is created
- Compare results between new 'correct' executable and 'buggy' installed version
- Did the patch fix the bug?

Testing!

- Test the following:
 - Modified ls works
 - Installed unmodified ls does NOT work
- Test on:
 - a file that has been recently modified
 - Make a change to an existing file or create a new file
 - a file that is at least a year old
 - `touch -t 201401210959.30 test_file`