Jon Parker
u0595680

**Testing Unix Patch (version 2.5)**

Let me preface this by saying I would not consider my testing procedure to be of much value.  I barely hit over half the code in patch, and my test files were very limited.  There were several command-line options that I couldn't even get working in my test code.  Furthermore, I had no oracle to use with patch.  Essentially my testing was a weak attempt at getting patch to crash without returning an error code (segfault, stack overflow, etc) and nothing bad ever occurred.  There were several times that patch returned an error code.  Passing in -c, -e, -u, and -R managed to crash with returned error codes but that seems expected because those are commands that tell patch that the specified files are special formats, and my files were not those formats.

My code coverage results are the following:

File 'patch.c'
Lines executed:61.67% of 600
Branches executed:63.00% of 546
Taken at least once:39.74% of 546
Calls executed:48.77% of 244

Obviously these are not stellar results.  In my defense, patch can take up to 30 different commandline args, almost all of which can be combined together.  Each argument has very specific interactions with the patch file being passed in.  This makes creating an oracle nigh impossible for someone who has never ever used Patch before.  Additionally, passing in many bad inputs results in patch popping up prompts where the user has to enter yes/no.  Since Patch takes diff files as it's input, one would also need to know how to use diff quite well to test patch since some of the diff formats that patch takes are a mystery to me.

Some cases I tried were using all of the possible command line args that didn't require a number or extra files, trying some files with weird unicode characters, trying some empty files, and passing in negative numbers into args that required number parameters.

Overall, I cannot with any confidence determine if Patch is solid.  I found no bugs, but my testing barely touches a fraction of what patch does.