Taylor Stapleton
chmod

For this test, I am using version 8.4, the standard chmod installed on the cade computers.

There are a lot of options on chmod. I went for what i thought would be the easiest break first. I downloaded and compiled (with quite some effort) the wisconsin shim tool called Murphy. Murphy helps you mock program system calls and access to common directories. Tests I performed included:

- causing all system calls to fail
- causing the tmp directory to appear full
- causing the CWD to appear extremely long
- causing reads to only read one byte at a time
- causing writes to only write one byte at a time
- adding delays to system calls returning

For the most part, chmod holds up really well against all of these tricks when used in a variety of situations and tests. There are however two scenarios where chmod doesn't handle its situation very gracefully.

1. When the call to Close a file descriptor fails
    - chmod exits with a negative exit status and prints a very unfriendly error message

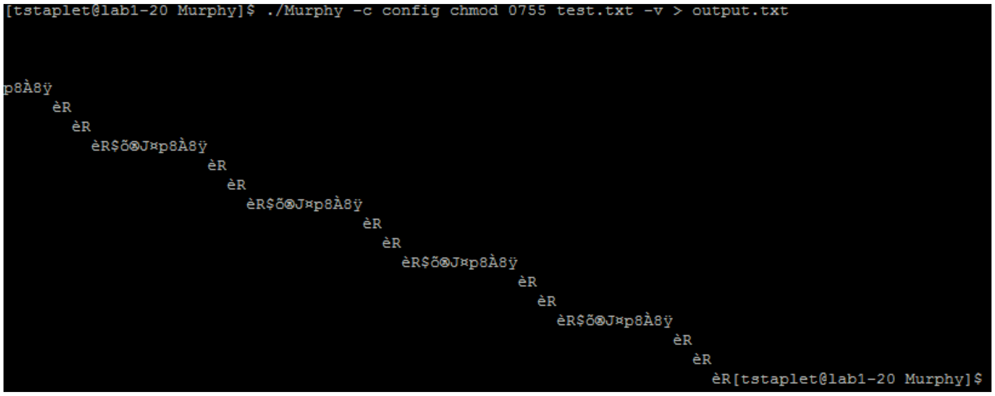
2. When used in -v(verbose) mode and we use the config line:
    - writezero:100

This config line tells Murphy to interrupt 100% of the write calls that chmod makes. Yeah, this is a difficult situation to handle for any program. But chmod has some strange behavior in this scenario.

- $./Murphy -c config chmod 0755 test.txt > output.txt
- In this scenario, chmod prints a bunch of garbage, certainly not the verbose information it is supposed to print.

Furthermore, we can also get some strange output using the same command, but with an invalid octal. The information printed for this second line makes it out to be a real reportable bug to the coreutils team.

Here is a particularly fun piece of output from
$./Murphy -c config chmod 0755 test.txt > output.txt

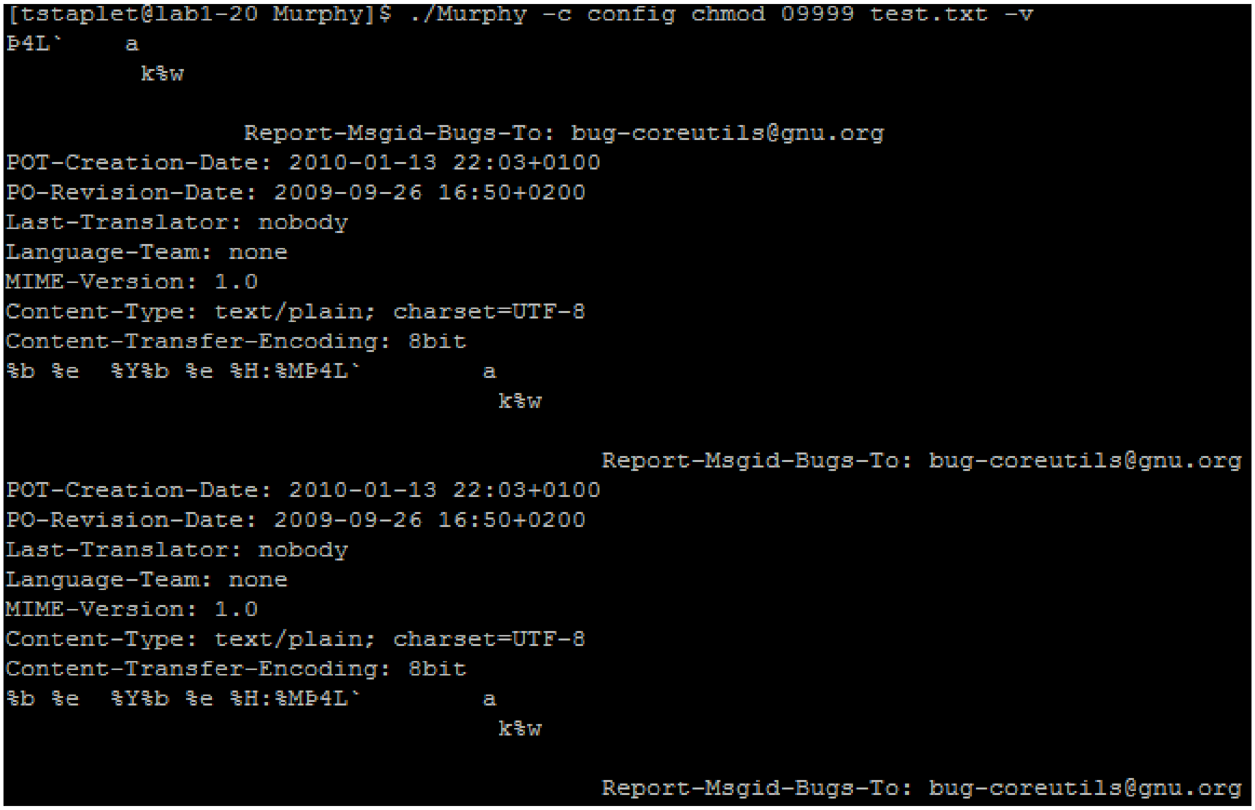


```
[tstaplet@lab1-20 Murphy]$ ./Murphy -c config chmod 0755 test.txt -v > output.txt



p8À8ÿ
     èR
       èR
         èR$õ®J¤p8À8ÿ
                     èR
                       èR
                         èR$õ®J¤p8À8ÿ
                                     èR
                                       èR
                                         èR$õ®J¤p8À8ÿ
                                                     èR
                                                       èR
                                                         èR$õ®J¤p8À8ÿ
                                                                     èR
                                                                       èR
                                                                         èR[tstaplet@lab1-20 Murphy]$
```

Here is part of the output from the invalid octal:
$./Murphy -c config chmod 09999 test.txt > output.txt



```
[tstaplet@lab1-20 Murphy]$ ./Murphy -c config chmod 09999 test.txt -v
Þ4L`    a
         k%w

                Report-Msgid-Bugs-To: bug-coreutils@gnu.org
POT-Creation-Date: 2010-01-13 22:03+0100
PO-Revision-Date: 2009-09-26 16:50+0200
Last-Translator: nobody
Language-Team: none
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8bit
%b %e  %Y%b %e %H:%MÞ4L`        a
                                 k%w

                                        Report-Msgid-Bugs-To: bug-coreutils@gnu.org
POT-Creation-Date: 2010-01-13 22:03+0100
PO-Revision-Date: 2009-09-26 16:50+0200
Last-Translator: nobody
Language-Team: none
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8bit
%b %e  %Y%b %e %H:%MÞ4L`        a
                                 k%w

                                        Report-Msgid-Bugs-To: bug-coreutils@gnu.org
```

This information was repeated in the output about 10 times.

Given the strange behavior in the above two scenarios, I cannot recommend that we rely on the functionality of chmod.