

Oscar Marshall

Testing Report for Cut

The *cut* utility is a part of *coreutils* which accepts a stream of characters over either *stdin* or a specified file and return a subset of characters according to the given arguments. *cut* has two modes of operation: characters (or bytes) and fields. In characters mode, *cut* will, for every line, print out only the characters in the columns specified. In fields mode, *cut* will, for every line, print out only the fields that are in the specified positions where a field is defined to be a group of characters that begins after a delimiter or at the beginning of the line, and ends at the next delimiter or at the end of the line. Although not as powerful as other text processing tools such as *awk*, *cut* is very efficient at extracting information from structured lines of text and leaves little room for error.

To test *cut*, I first created a suite of test cases which covered all of the possible argument combinations which would produce different results. When none of these cases crashed the program or produced unexpected results, I began feeding inputs that lay just outside of the operating boundary to make sure *cut* would print an error and quit gracefully. I then rebuilt *cut* using the *-fsanitize=undefined* flag to make sure that none of the cases executed any undefined behaviour. When no undefined behaviour cropped up, I ran *cut* under *valgrind* to make sure that there existed no memory leaks. Looking at the code and failing to find any bugs or unhandled cases, I fed *cut* itself as the argument to make sure that it wouldn't crash when processing a file formatted for a completely different use.

Although I was unable to crash *cut*, there is always more testing that can be done. So far, my testing has led me to believe that *cut* is a solid piece of code, and reading over the code has convinced me of its correctness. There are still more involved tests that could be run which cover more corner cases and different operating conditions, but based on what I've observed so far, I believe that *cut*, in fact, does what it claims to do.