Mark Randles
Trevor Reiter
2008-01-20

Our group chose to look at a piece of software, the pico text edtiort. Our analysis would also apply to it's direct clone nano. The pico text editor was originally provided as part of the pine email package, but was used for many other applications besides email. There are a number of design concepts in the pine interface. First and foremost, it was to be a lightweight text editor. There was to be no complex plugin system, no syntax parsing, no integration with other applications. There are a limited number of commands in the pico editor. The second design concept was that it should work from a command-line in almost any terminal.

The primary users of pico were originally pine users, so any user who was using a Unix system for email and possibly other purposes. The original user group would have had some command-line savvy, but it was not required they be a Unix power-user. Over time, due to pico's ease-of-use, it became the defacto text editor for people new to the Unix environment. Compared to vi and emacs, two other Unix text editors, pico is closer to a "word-processing" experience in command controls then the other two, hence lowering the knowledge curve required to learn to use the text editor.

There are two primary use scenarios for the pico text editor. The first is obvious, that of a text editor. The user should be able to edit and save an existing or new text file. The second use scenario is that of a text viewer. In this mode, pico allows for the viewing of text but does not allow the user to make or save any changes.

The interaction techniques in pico are relatively simple. Since there is a limited number of similar use cases all these techniques can be applied to both. The most important interaction technique is that of text entry. By default, unless a special key is held all keyboard input is inserted into the document wherever the cursor is positioned. The second interaction technique is that of command sequences. To perform a special action, like saving a file a two key special character is input from the keyboard. Usually this is a standard Unix escape character, which is captured by the pico editor. Some sequence may require a second step, such as saving a file requires the user to hit enter or type a new file name, and these steps are show by replacing the bottom of the terminal with a context sensitive menu.

The actual implementation of pico requires no futher technology then what would be required by a contemporary Unix system in the middle 1980's. This includes terminals that were capable of sending signals and special characters and which provided ways to move a cursor in the terminal. Pico also requires the ability for text input/output at arbitrary portions of the terminal screen.

The user of pico requires only the barest minimum of experience with other text editors and the Unix system to use pico. They would require the knowledge that hitting the "ctrl" key with another key would trigger the appropriate special action. However, all of the special actions are listed, usually, at the bottom of the screen. All special actions that require a second step also provide on-screen help and are clearly marked. Only enough command-line skill is required to invoke the software.