Text Files and Text Editors

OPS102 Week 3 Class 2

Tiayyba Riaz/John Sellens

January 21, 2025

Seneca Polytechnic

Outline

What is a Text File?

Text Editors

What is a Text File?

Files, Files, Files

- Last class we mentioned:
- Data is saved in files
- In Linux/Unix we *really* like text files
- For data, presentations, configuration, logs, and more
- The system and shell provide "easy" ways to deal with files
- More about file details next week

Line Endings in Text Files

- Each line in a text file as some end-of-line indicator.
  - In UNIX/Linux it's "new line" (line feed or ^J or \n).
  - In Windows, it's "carriage return" (^M or \r) followed "line feed".
  - In MacOS, it's "carriage return" (^M or \r).
- When text files are transferred between systems, there could be problems.
  - A file moved from Windows to Linux might show some empty lines or ^M's
  - A file moved from Linux to Windows may miss the line breaks.
- Some text editors notice and try to help you out.
- See the `dos2unix(1)` and `unix2dos(1)` commands.
- Remember how manual typewriters and teletype machines worked?

Text Editors

Text Editors

- A text editor allows users to create, modify and save editing changes of text files.
- Editing text files is an everyday activity for both programmers as well as administrators on a Unix and Linux system.
  - Edit System configuration files
  - Write Scripts and programs
  - Write/edit documentation
  - Develop web pages
- It is important to learn to be able to user a text editor in order to install, configure and run network services.

Text Editors – History

- Before there were video (CRT) terminals, when paper terminals were the rule, "line editors" were used
  - e.g. qed, ed, edlin (DOS)
- On Unix, the `ed(1)` editor was replaced by `ex(1)` (short for EXtended)
  - https://en.wikipedia.org/wiki/Ex_(text_editor)
- And then a "visual mode" for ex was created, and the `vi(1)` editor was born.
  - https://en.wikipedia.org/wiki/Vi
- ex and vi both came from Bill Joy at UCBerkeley and appeared first in BSD.
- Syntax that originated in `ed(1)` is still part of `vi(1)`, `sed(1)`, and other tools.
- The `grep(1)` command echoes the ed syntax of: `g/regexp/p`

Text Editors – Many Choices Today

- There are a number of editors available in Linux and Windows.
- Try several, choose the one that's best for you.
- Examples of some common and well known text editors:
    - VIM
    - Nano
    - Gedit
    - Notepad
    - Notepad++
    - Sublime
- See also: https://en.wikipedia.org/wiki/Comparison_of_text_editors
- You will learn some basics of the Nano editor in next slides. Nano is the default text editor for many Linux distributions.

Nano Text Editor Introduction

- The Nano text editor is an easy to use text editor.
  - Installed on many/most, but not all, Linux systems.
- Run the command **nano** and the editor will open for you.
- Nano editing commands typically control characters.
  - For example, the notation **^G** means press and hold the "control" key on the keyboard, then press the G key.
- For nano you don't need to memorize many commands, since the menu is always at the bottom of the nano screen.

More About Nano

Some basic commands include:

- ^G – to get help at any time
- ^R – open a file
- ^O – save a file
- ^W – find
- ^\ – replace
- ^X – exit
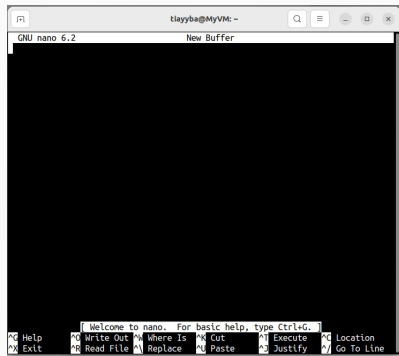
Nano Editor Start Up Screen

Nano vs Vi

- These days, the `vi(1)` command is often actually `vim(1)`.
    - `vim(1)` adds many more modern features to classic vi.
    - Many intended specifically for programmers.
- For casual, occasional use, the nano editor is fine.
- For anything beyond the absolute basics, you really should become familiar with `vi` (or `vim`)
    - Check out the `vimtutor(1)` command to get started.

Summary

- Text files are good
- Beware of different line endlings on different OSs
- Be familiar with Nano and Vi
- The vi/vim editors are very useful
    - Though modern IDEs offer many features