

Introduction

OPS102 Week 1 Class 2

Tiayyba Riaz/Chris Tyler

September 3, 2024

Seneca Polytechnic

Outline

Recap From Last Class

A Bit of History cont'd

Command Line and Commands

Open Source and Proprietary Software

Recap From Last Class

Recap From Last Class

- Course introduction and overview
- Operating system basics, components, and interfaces
- A bit of history

A Bit of History cont'd

A Bit of History cont'd

- Recall we talked a little bit about the history of UNIX
- And PC-DOS / MS-DOS
- And PC-DOS / MS-DOS 2.0 added hierarchical filesystems
 - With backslash as the pathname separator

Operating Systems get GUIs

- GUIs were added to Unix systems starting in 1984 with the introduction of the *X Window System* from Project Athena at MIT.
- DOS received multi-tasking abilities and a GUI with the introduction of *Windows 1.0* in 1985.
- *Windows NT 3.1*, released in 1993, was a new 32-bit operating system that combined features from DOS/Windows and a jointly-developed Microsoft-IBM operating system named *OS/2*.
 - Not based on MS-DOS, but compatible with MS-DOS
 - All modern versions of Windows are derived from Windows NT.
- Other early GUI systems include Xerox Star (1981), Apollo DN100 (1981), Sun Sun-1 (1982), Apple Lisa (1983), Apple Macintosh (1984)

Unix Evolves from Product to Standard

- Unix was originally a proprietary product of Bell Labs / AT&T.
- Eventually, it became a family of related operating systems customized by various vendors and sold under many different names (XENIX, AIX, Ultrix, HP/UX, and more).
- Various standard bodies, including the IEEE, The Open Group, and ISO created standards to define what a “Unix-like” operating system should contain and how it should operate.
- Ultimately, Unix became a trademark rather than a product, and any operating system that was tested and proven to conform to the Unix standards was granted permission to use the Unix trademark.

- While originally considered valuable, the importance of the Unix trademark is no longer as significant as it once was.
- Many contemporary operating systems are regarded as Unix-like, although few have gone through official conformance testing.
 - Apple's MacOS is officially a Unix system
 - Linux is not officially a Unix system

Unix-Like Operating Systems

- Linux
- Mac OS and iOS
 - Though generally only “under the covers”
- FreeBSD and OpenBSD and NetBSD
- And many others!

Operating Systems In this Course

- We're going to focus on two operating systems:
 - Windows (version 10), a common proprietary operating system
 - Linux, representing Unix-like operating systems
- Windows 10 is installed on the lab computers.
- A Linux CLI is available by remote connection (through a tool called SSH) to a large server system called Matrix.

Command Line and Commands

- A CLI is provided by two components:
 - A terminal program which accepts user input from a keyboard and displays text output.
 - A shell which interprets commands, starts and stops programs, displays the output of those programs, and enables the user to interact with those programs.
- These programs are separate so they can be mixed-and-matched for different contexts.

- Bash – a shell widely used on Linux systems, also available on Windows
- PowerShell – a shell widely used on Windows systems, also available on Linux
- CMD – a shell available only on Windows systems
- In this course, we're going to focus on Bash and CMD.

What is a Command?

- A small number of commands are built into the shell
- All other commands are separate programs, each contained in its own file.
- Any GUI or CLI program, including things like the Chrome web browser, git version control tool, or the LibreOffice Writer word processor are also commands!
- Commands may also be called programs, utilities, applications, or executables – the terms are largely interchangeable.

Command Structure

- Most commands have a similar structure:

`command_name arguments ...`

- "`command_name`" is the name of a built-in command, or a separate external program.
- "`arguments ...`" are zero or more tokens for the command, which could be *options*, or *values* for the command to use

Type of Arguments

- Option (also called a Switch or Control Argument) – these alter the operation of the command
- Option with a Value – as the name indicates, these consist of an option immediately followed by a value
- Positional Arguments – these are all other arguments, and their position (sequence) may be significant

Options – Examples

```
ls          # Linux command to list files
ls -l      # The -l option selects "long" output

dir        # Windows command to list files
dir /b     # The /b option selects "bare" output
```

Options – Long and Short Versions (Linux)

```
ls -l -h           # Show human-readable sizes
ls -l --human-readable # Same thing
ls -lh            # Same thing
```

Option with a Value

```
ls -l -sort=size # Show files sorted by size (Linux)
```

```
ls -l -sort time # Show files sorted by time (Linux)
```

```
dir -o:s # Show files sorted by size (Win10)
```

```
dir -o t # Show files sorted by time (Win10)
```

```
dir /o:s # Show files sorted by size (Win10)
```

```
dir /o t # Show files sorted by time (Win10)
```

Positional Arguments

```
cp old new    # Copy file "old" to "new" (Linux)
```

```
copy old new  # Copy file "old" to "new" (Win10)
```

Open Source and Proprietary Software

- Software is generally protected by copyright, an international legal mechanism that controls how written works may be used.
- The party which controls these works – either the employer of the author, or the author themselves if outside of an employment context – is the copyright holder.
- The copyright holder can license the software to another party.

Open Source and Proprietary Software

- There are two broad categories of software licenses in use:
- Proprietary licenses – the copyright holder provides the software to other parties on a limited basis, often for a fee. The human-readable source code for the software is usually not provided.
- Open Source Licenses – the copyright holder permits other parties to use the software under one of the licenses that conform to the *Open Source Definition* (<https://opensource.org/osd/>). This enables other parties to alter, adapt, change, rebuild, use, and sell the software, subject to certain restrictions.

Types of Open Source Licenses

- Open Source Licenses are further divided into two categories
- Free Software Licenses – these licenses require anyone redistributing the software to do so under the same terms as the software was received.
 - (<https://www.gnu.org/philosophy/free-sw.html.en>)
- Permissive Licenses – these licenses allow the open source software to be redistributed without the same permissions as the software was received under.
- What are the advantages and disadvantages of each approach?

End of Week One!

- Lots of new ideas and concepts to digest
- You likely haven't use the command line much (if at all!)
- Lots more to come!