Chapter 7: Finding Files

The locate Command

- Useful for fast searches of a filesystem for a file(s)
- Uses a database which is updated once a day
- Database updated by administrator using the `updatedb` command
- Syntax:

  $locate filename

The locate Command

- The `locate` command only will return results of files that the current user would normally have access to
- The `locate` command will display all files that has the search term anywhere in the file name
- The `locate` command is case sensitive

The locate Command

- Advantages
  - Fast. Searches a database of all files on the computer verses the live filesystem the find command uses
- Disadvantages:
  - The database is only updated once per day so new files are not in the database
  - You can only search for files by name verses other search criteria the find command supports

The find Command

- Searches a live filesystem for specified file(s)
- Supports different search criteria options
- Searches the live filesystem which can take a large amount of time
- Slower than the locate command
- Example: `find / -name file_name(s)`

The find Command Options

Example	Meaning
-iname LOSTFILE	Case insensitive search by name
-mtime -3	Files modified less than three days ago
-mmin -10	Files modified less than ten minutes ago
-size +1M	Files larger than one megabyte
-user joe	Files owned by the user joe
-nouser	Files not owned by any user
-empty	Files that are empty
-type d	Files that are directory files
-maxdepth 1	Do not use recursion to enter subdirectories; only search primary directory

The whereis Command

- Displays the directory location and man page for the specified command
- Searches only the directories defined by the $PATH variable
- Example:

  $ which grep
  grep: /bin/grep /usr/share/man/man1/grep.1.gz

The which Command

- Displays the directory location(s) of the specified command or script
- Returns the location of the real command
- Searches only the directories defined by the $PATH variable
- The -a option is used to locate multiple executable files. Useful to know if an executable script was inserted maliciously to override an existing command.

Introduction to FHS

- The Filesystem Hierarchy Standard (FHS) is standard that specifies standard directories and their content for use with a filesystem
- Helps to know what directories to expect to find and what to find in them
- Allows programmers to write programs that will be able to work across a wide variety of systems that conform to this standard

History of FHS

- First known as known as the Filesystem Standard (FSSTND)
- Renamed in 1997 with series 2
- Final series 2 (2.3) published in 2004
- Draft version of series 3 published in 2001
  - Still under review
  - Unofficial changes ongoing

Recent unofficial changes

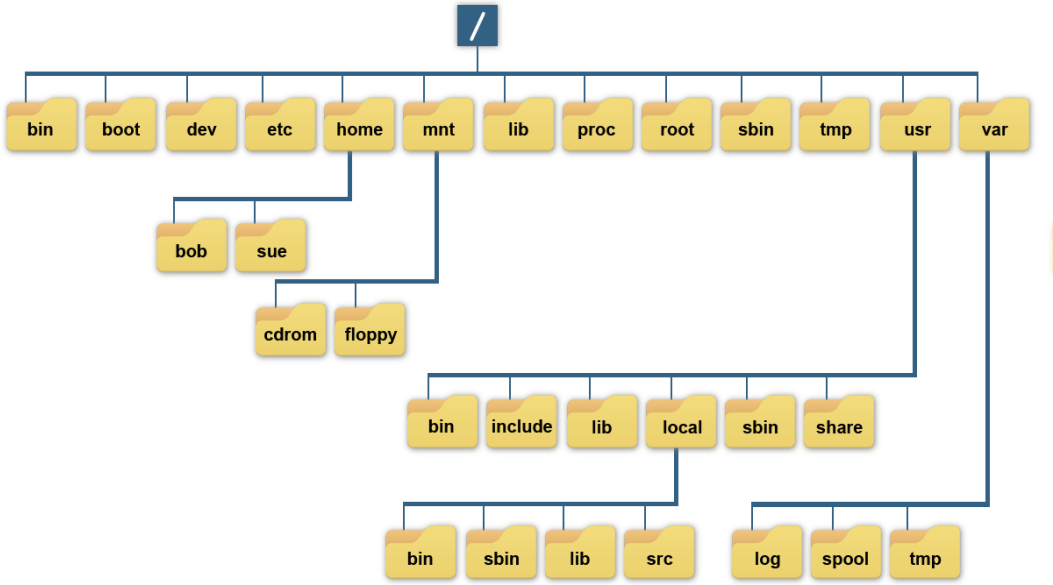
- New directory additions:
  - `/run` to contain volatile data that changes at runtime
  - `/sys` to hold files related to the kernel
- Directory merging:
  - `/bin` merged into `/usr/bin`
  - `/sbin` merged into `/usr/sbin`
  - `/lib` merged into `/usr/lib`

Directory classifications

- If a directory structure is classified as *shareable*, then it typically does not contain anything that would be unique to a particular system like a configuration file
- If a directory structure is classified as *static* it means that it usually doesn't change and may suggest that it might be mounted read-only
- If a directory structure is classified as *variable*, it is likely to change and would have to be available for both read and writes

Important directories

- FHS standard details many directories
- Administrators should know those on the next three slides



Important directories

Directory	Purpose
/	The root of the primary filesystem hierarchy
/bin	Contain essential user executables
/boot	Contain the kernel and bootloader files
/dev	Populated with files representing attached devices
/etc	Configuration files specific to the host
/home	Common location for user home directories
/lib	Essential libraries to support /bin and /sbin executables
/mnt	Mount point for temporarily mounting a filesystem
/opt	Optional third party add-on software
/root	Home directory for the root user
/sbin	Contains system or administrative executables

Important directories (cont.)

Directory	Purpose
/srv	May contain data provided by services of the system
/tmp	Location for creating temporary files
/usr	The root of the secondary filesystem hierarchy
/usr/bin	Contains the majority of the user commands
/usr/include	Header files for compiling C-based software
/usr/lib	Shared libraries to support /usr/bin and /usr/sbin
/usr/local	The root of the third filesystem hierarchy for local software
/usr/sbin	Non-vital system or administrative executables
/usr/share	Location for architecturally-independent data files
/usr/share/dict	Word lists

Important directories (cont.)

Directory	Purpose
/usr/share/doc	Documentation for software packages
/usr/share/info	Information pages for software packages
/usr/share/locale	Locale information
/usr/share/man	Location for man pages
/usr/share/nls	Native language support files