CSCI 274 - Intro to Linux OS

Week 4 - I/O Manipulation, Three Data Streams, Redirection, Linux Waste Bin and Variables

Rocco Marchitto (rmarchitto@mines.edu) [A/B/D]
Colin Randall (crandall@mines.edu) [C]

Overview

1. I/O Manipulation
2. Three Data Streams
3. Redirection
4. Linux Waste Bin
5. Variables

I/O Manipulation

echo - used to display line of text that is passed as an argument

$ echo [-e | -n] [TEXT]

cat - (con**cat**enate) reads data from the file gives their content as output. Helps create, view, and concatenate files.

$ cat [-n] ... [FILE] ...

Three Data Streams

Data streams are objects used to transfer information around the shell. Standard in, out, and error are file references to data streams.

- stdin = standard input stream; keyboard by default
- stdout = standard output stream; terminal by default
- stderr = standard output for error messages; terminal by default

Type	Symbol
stdin	0<
stdout	1>
stderr	2>

Redirection

Most Unix system commands take input from your terminal and send the resulting output back to your terminal. The output from a command can be easily diverted to a file instead.

- command > file = output of the command will be written to the file instead of your terminal
- command >> file = will append the output in an existing file

> or 1> for standard output

Type	Symbol
stdin	0<
stdout	1>
stderr	2>

Redirection

A command normally reads its input from the standard input, which happens to be your terminal by default.

The input of some (not all) commands can be redirected from a file.

- command **<** file = input can be redirected from a file in this manner

  < or **0<** for standard input

Type	Symbol
stdin	0<
stdout	1>
stderr	2>

Redirection

By default, errors are treated the same as stdout. It sends them directly to your terminal screen. Error messages can be filtered to their own output file.

- command **2>** fileError = can filter out the error messages from a command result and save them to a file
- command **2>>** fileError = can filter out the error messages from a command and append them to an existing file

standard error must use **2>/2>>**

Type	Symbol
stdin	0<
stdout	1>
stderr	2>

Linux Waste Bin

Perhaps we're not interested in output messages (default is going to the terminal) and you wish to discard them. In this case, you can redirect into a special file on the Linux system called **/dev/null**.

/dev/null is similar to the "Recycle Bin" under Windows except it's a waste paper basket with a point of no return - the Linux black hole! Once information has gone into /dev/null, it's gone forever.

Example: find / -name bin 2> /dev/null

To find all “bin” files in the home directory recursively, but avoid printing errors for any “bin” files that do not allow access.

Variables

- Environment variables are those that define system properties and allow programs to function correctly.
- Local variables are only present in the current instance or session of the shell.

Setting a variable: **var_name=”value”**

Use **echo** with a dollar sign to print the value of variables: **echo $var_name**

export - marks an environment variable to be exported with any new program/script and thus it allows a program/script to inherit all marked variables.

$ export [-n | -p] [name[=value] ...]

Due Dates

- Week 4 Quiz - Sep 18
- UnixHistory.sh - Sep 25