

# CIS241

## System-Level Programming and Utilities

### Redirection and Piping

Erik Fredericks, [frederer@gvsu.edu](mailto:frederer@gvsu.edu)

Fall 2025

Based on material provided by Erin Carrier, Austin Ferguson, and Katherine Bowers



# First, command chaining

Or, running multiple commands in a single line

Seperate with `;` or `&&`

- `date ; who ; whoami ; echo "Hi there!"`
- `date && who && whoami && echo "Hi there!"`
  - Don't use a single `&` by mistake - that means run in the background!

# A reference

<https://dev.to/bowmanjd/bash-execution-tips-for-shell-jockeys-and-script-fabricators-5dan>

Use `||` to execute one command only when the previous one fails.

Combine the above for conditional branching.

Use `;` to join two commands when you want the second to execute no matter the result of the first one.

Use `&` to run the first job in the background while the next executes. Follow all with wait for a clean return to the command prompt

# IO redirection

## Standard streams

- 0: `stdin` - Standard input (user typing)
- 1: `stdout` - Standard output (print to terminal // output of program)
- 2: `stderr` - Standard error (also prints to terminal)

# IO redirection

Redirect `stdout` to file:

- `ls -la > dir_contents`

Redirect `stderr` to file:

- `ls -la 2> errors`

Redirect file to `stdin`:

- `sort < my_file.txt`



# Example:

Redirect `stdout` to file and `stderr` to stdout:

- `command >output 2>&1`

Redirect `stdout` to file and `stderr` to another file:

- `command >output 2>output.err`

Redirect `stderr` to the void:

- `command 2> /dev/null`

`command` = `echo "Hi"` or `ecccccho "Hi"`

Command	Description
command 1>file command >file	The Standard Output of the command is sent to a file instead of to the terminal screen.
command 2>file	The Standard Error of the command is sent to a file instead of to the terminal screen.
command 1>fileA 2>fileB command >fileA 2>fileB	The Standard Output of the command is sent to fileA instead of to the terminal screen, and the Standard Error of the command is sent to fileB instead of to the terminal screen.
command 1>file 2>&1 command >file 2>&1 command 1>&2 2>file com- mand >&2 2>file	Both the Standard Output and the Standard Error are sent to the same file instead of to the terminal screen.
command 1>>file command >>file	The Standard Output of the command is appended to a file instead of being sent to the terminal screen.
command 2>>file	The Standard Error of the command is appended to a file instead of being sent to the terminal screen.
command 0<file command <file	The Standard Input of a command is taken from a file.

Table 7-1: Common redirection examples



# Piping



# Piping

Send output of one command as input to another

```
command1 args | command2
```

- `cat file | sort`
- `cat file | sort | uniq`
- `cat file | sort | uniq > output.txt`

