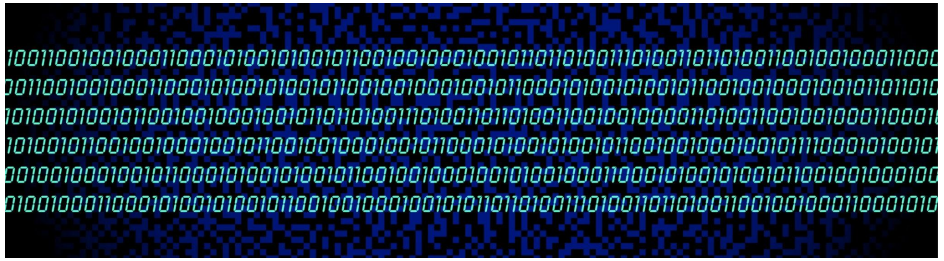
Redirection and Pipes

CS14 - S26

Redirection and Piping allow us to…

- Quickly save output of commands
- Feed large inputs to commands
- Write commands that combine and filter data
- Combine commands to make complex data streams

Data Streams



- Information is stored as bits
- Abstractly, information is sent between parts of the computer in linear “streams” of bits.
- File stream – break a file down to bits and send each bit one by one
- Input/Output stream – bits going to or coming from somewhere

- The terminal recognizes three “standard streams” and can read/write data to them
  - stdin
  - stdout
  - stderr

stdin, file descriptor 0 

- Stream from terminal to a program as input data.
  - e.g. input() in Python, cin in C++, scanf() in C
- A terminal command can be given stdin input from a file using the stdin redirection operator: <
  - Useful for programs that expect input from standard input
  - Some utility commands work differently based on using standard input
- e.g. wc file vs streamed file
  - `wc blah.txt`
  - `wc < blah.txt`

stdout, file descriptor 1 >

- Stream from program that is printed to the terminal.
- Output normally printed to the terminal can be saved to a file instead using the stdout redirection operator: >
  - Useful to save output of programs that print to terminal, or save result of commands
  - Can be used to discard output: > /dev/null
- e.g. save results of wc
  - `wc blah.txt`
  - `wc blah.txt > blah_stats.txt`
- Default behavior is to overwrite! 
- Can instead append using >>

stderr, file descriptor 2 &>

- Stream that is used to report error information. By default, it is printed to the terminal.
- Information in stderr can be saved to a file instead using the stdout redirection operator in conjunction with stderr to stdout redirection: &>
  - Useful to log error messages
  - Can discard output and only save error messages, but the syntax is ugly
  - In the wild, may also see equivalent expression 2>&1. Which is supported is shell dependent.
- e.g. save output and errors of ls
  - `ls real_file fake_file` (produces output and error message)
  - `ls real_file fake_file > ls_results` (still prints error message to terminal)
  - `ls real_file fake_file &> ls_results` (because of redirection, saves error message)
- e.g. save only errors of ls
  - `ls real_file fake_file &> ls_errors 1>/dev/null`

Pipes

- Use stdout of one operation as stdin of a second operation
  - NOTE: The pipe symbol is Shift+\ on typical keyboards, not capital `I` or lowercase `l`.
- Can use pipes to make complex data operations from simple commands
- e.g.
  - `echo “I love computer science!” | wc`
  - `ls -1r / | head -n5 > first_five`

New utilities that work with stdout/stdin

- cut
  - Only output (“cut”) parts of each line of input
- uniq
  - Filter/modify adjacent, duplicate lines in input
- sort
  - Sort lines of input
- grep
  - Search for text in input

Practice using these commands with pipes following the instructions in:

`/home/ckazer/public/cs14/inclass/pipes_exercise.md`