

Globbering, pattern matching

- Globbering is the term used for bash's form of pattern matching in commands
- It is used when we want to use a pattern to describe a set of strings, e.g. all filenames ending in .c, all directories that have a digit as the third character, etc
- When the user types a command that includes globbering characters, bash figures out all the matches (if any) before passing the strings along to the program/command using them

The * and ? wildcards

- The * matches 0 or more characters, so if we type a command like “ls *.cpp” it will match all filenames (in the current directory) that end in .cpp
- The ? matches any single character, so if we type a command like “ls foo?blah” it will match all filenames (in the current directory) that start with foo, then have another character, then finish with blah

Pattern choices

- You can specify you want matches for any one of a set of patterns using { }
- e.g. `ls {r*.c,f??}` would match any file beginning with r and ending with .c OR any file with a three-character name that begins with f
- Note that bash will yell at you if you have whitespace after the { or before the }

Specifying a set of characters, []

- We can use syntax like [xyz] to specify the character we want can be any of the ones inside the square brackets, x, y, or z in this case.
- We can also specify ranges, e.g. [a..z] matches any character from a to z
- The ^ can be used to invert this, specifying anything except the characters listed, e.g. [^1..9] means anything except the digits 1 through 9

Example

- Suppose we want to list all the files (in the current directory) that start with an R and end in a digit followed by a two-character extension
- Is $R^*[0-9].??$