

**CSE 384 LECTURES**

**YUZHE (RICHARD) TANG**

# LECTURE 3: FIND AND GREP

# REFERENCE

- "Bash Guide for Beginners" [[link](#)]

# FILENAME EXPANSION

- Expansion: a "pattern" string to match a specified class of file names
- Syntax: Special characters
  - \* matching a string of an arbitrary number of any characters
  - [ abc ] matching one character, either it is a, b or c
- Examples:
  - `ls ./ * .pdf #in shared folder`
  - `ls ./Lecture* .pdf`
  - `ls ./Lecture[12]* .pdf`
  - `ls ./Lecture[34]* .pdf`

# FIND

1. `find` is a command you can use to find files by filename, type, etc.

# FIND: FORMAT AND FLAGS

1. Basic command: `find . -name file_a`
2. Flag `-name`: find files by matching file name (or name expansion)
  - `find . -name "hello.c"`
3. Flag `-maxdepth`: the depth of directories to the find command should go
4. Flag `-type d`: target is to find directories, not files
  - `find / -maxdepth 1 -type d`

# ADVANCED FIND

- Use filename expansion for a more powerful `find`:
  - `find . -name "*.c"`
  - `find . -name "*.pdf" #in shared folder`
  - `find . -name "Lecture*.pdf"`
  - `find . -name "Lecture[12]*.pdf"`
  - `find . -name "Lecture[34]*.pdf"`

# GREP

- `find` command cannot find files by file content
- `grep` command: search file content
  - like a little Google service running your Linux file system.



# GREP: FORMAT AND FLAGS

1. Basic command: `grep search_term path`
  - `path` can be a path to a file or a directory
  - Example: `grep hello hello.c`
2. Flag `-r`: search files recursively under a given directory.
  - Example: `grep -r hello .`
3. Flag `-i`: the search term is case-insensitive
  - Example: `grep -i HELLO hello.c` (it matches for both HELLO and hello)

# ADVANCED GREP

- `grep` is a string-matching problem:
  - `grep term filename` is to match `term` with each text line in file `filename`.
  - You can write more complex "term" or **pattern** to do powerful content search

# REGULAR EXPRESSION

- The language for writing grep pattern is called **regular expression** or regex
  - What filename expansion is to `find` is what regex to `grep`
  - `expansion` specifies the search pattern for filenames in `find`
  - `regex` specifies the search pattern for file contents in `grep`
  - Regex has a different syntax from expansion!

# REGEX: STRING-MATCHING PROBLEM

- match: a grep in a file decides the match of the pattern to all lines in the file.
  - `grep Alice file_a` (assume two lines in `file_a`)
  - `match(Alice, line 1 in file_a) = 0/1`
  - `match(Alice, line 2 in file_a) = 0/1`
- Example: `match(Alice, Alice) = 1`
- Example: `match(Alice, Bob) = 0`

# REGEX SYNTAX (1)

1. asterisk \*: matching previous character repeating arbitrary times (including zero time).
  - `match(1133*, 113)=1`
  - demo: `echo 113 | grep 1133*`
  - `match(p, s)` decides if there is any substrings in `s` that match pattern `p`
2. dot .: matching arbitrary single character
  - `match(13., 13)=0`
  - `match(13., 134)=1`

## REGEX SYNTAX (2)

3. Brackets `[ . . . ]`: enclose a set of characters

- `match(1[345],13)=1,match(1[345],15)=1,match(1[345],18)=0`
- `match(1[3-5],14)=1`
- `match(1[^3-5],14)=0,match(1[^3-5],18)=1`

4. caret `^`: beginning of a line

5. dollar sign `$`: end of a line

- `^$` matches blank lines.

# EXERCISE 1

- **Setup:** Follow instructions below to create a file `file1`
  - `touch file1`
  - `echo "hello too" >> file1`
- Try the `grep` commands below. Report what's shown in your prompt to BB.
  - `grep ^hello file1`
  - `grep hello$ file1`
  - `grep t[wo]o file1`
  - `grep ^[A-Z] file1`

# EXERCISE 2

- **Setup:** Follow instructions below to create needed directories and files
  - `mkdir -p exec2/dir1; mkdir -p exec2/dir2`
  - `cd exec2; echo "hello world" >> a_file`
  - `echo "world hello" >> dir1/file_a`
  - `echo "hello" >> dir2/file_b`
  - `echo "yahoo" >> dir2/file_b`
- 1. Design a command to find all the files of name starting with `fil` under directory `exec2`.
  - Submit to BB both your command and terminal prompt.
- 2. Design a command to find all the file lines under directory `exec2` that contain a *single word* "hello".
  - Submit to BB both your command and terminal prompt.