

# CS 136L Lecture 3

## More Linux Shell

## Globber Patterns

- Wildcard matching - What do these symbols do:

? \* [] [!] { }

- Warning! What do each of these commands do?

- `echo "*.txt"`

- `echo '*.txt'`

- `echo *.txt`

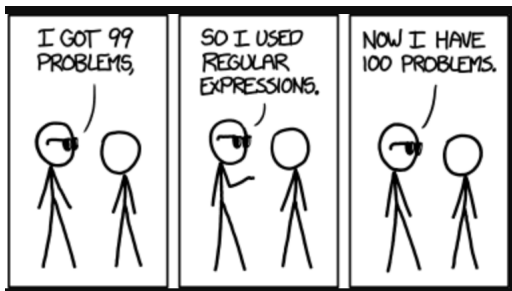
- `find . -name *.txt`

- `find . -name "*.txt"`

# Regular Expressions with egrep

- When would you use egrep?
- Symbols: [] [^] . | ( ) ? \* + ^ \$
- Pitfall: **Egrep patterns may differ from globbing patterns!**
- Pitfall: How to I search for a special character in a string like a . or a ? in a file?
- Pitfall: Be careful about ^ and \$!

# Regular Expressions



Source: [https://imgs.xkcd.com/comics/perl\\_problems.png](https://imgs.xkcd.com/comics/perl_problems.png)

# Piping

- What does the following do?

```
ls *.c > output.txt
```

```
wc -l < output.txt
```

- How can we do this without creating the intermediary file?

## Standard Input vs Argument to a Program

A forgetful programmer places the following in `path_dict.txt`:

```
/usr/share/dict/words
```

How can we print the contents of the above file via the text file?

## Standard Input vs Argument to a Program

A forgetful programmer places the following in `path_dict.txt`:

```
/usr/share/dict/words
```

How can we print the contents of the above file via the text file?

What do each of these commands do?

- `cat path_dict.txt`
- `cat < path_dict.txt,`
- `cat $(cat path_dict.txt)`

# Running Files

- What if you wanted to execute the contents of a file?



# Running Files

- What if you wanted to execute the contents of a file?
- We saw this in module 0!

```
source <filename>
```

# Permissions

- Try `ls -l`. What information do you see displayed?
- `chmod a+x <filename>` add executable permission to file for all users.
- `chmod a+rwx <filename>` add read, write and executable permission to file for all users. This is a bad idea - why?
- What is the numeric equivalent to the above?

## Warnings About Permissions



Source: <https://memegenerator.net/instance/68424931>