**Cygwin and "filenames that contain spaces"**

**Problem:** Search every file in a file hierarchy and find every email address stored in the files.

Using Cygwin, to access your Windows 7 file hierarchy:

cd /cygdrive/c

To deal with file names that contains spaces:

* The spaces have to be escaped: This\ is\ an\ example
* Put the name inside single quote marks: 'This is an example'  
  (The string is used exactly as it is. Single quotes (') can't appear in the string.)
* Put the name inside double quote marks: "This is an example"

(Shell expansions like $(cmd) and $variable are done inside the string before it is used.)

Using find, here are several possible techniques to quote filenames:

* Terminate file names with a '\0' character; then tell xargs that lines are '\0' terminated:

find . –print0 | xargs -0 cmd

* Use the –printf flag for find to print the file name with quotes:

find . –printf "\"%p\" \n" | xargs cmd

find . –printf "'%p' \n" | xargs cmd

* Use sed to add quotes around each file name

find . -type f | sed -e "s/^/'/;s/$/'/" | xargs cmd

By default, "white space" (spaces, tabs, new-lines) separates "words" in a string.

names='alpha beta gamma'

for n in $names; do

printf "$n\n"

done

IFS stands for "internal field separator" and its value is used to separate "words" in a string.

IFS=$'\n'

names='alpha\nbeta\ngamma'

for n in $names; do

printf "$n\n"

done

The =~ comparison operator performs *regular expression pattern matching*: (Does the variable contain the pattern?

if [[ "$var" =~ "pattern" ]]; then

Here are some possible regular expressions for matching email addresses:

"\w+(\.\w\*)\*@\w+(\.\w+)\*"

"\w{1,12}(\.\w{0,12}){0,10}@\w{1,12}(\.\w{1,12]){0,8}"