CA170: Week 8

Search Engine Shell

How to write a search engine in 9 lines of Shell

- The following is a search engine for a website in 9 lines of Shell:

```
#!/bin/sh

echo "Content-type: text/html"
echo

echo '<html> <head> <title> Search results </title> </head> <body>'

argument=`echo "$QUERY_STRING" | sed "s|q=||"`

cd /users/homes/me/public_html

echo '<pre>'
grep -i "$argument" *html */*html        |    sed -e 's|<|\&lt;|g'    |    sed -e
's|>|\&gt;|g'
echo '</pre>'
```

Notes

1. his is an online program. It is a server-side CGI script.
   It accepts input through a HTML form.
2. `"q="` assumes that your input variable is called `"q"` in the HTML form.
3. Your web directories need to be readable for the wildcard to work.
4. We pipe the result of grep into an ugly-looking sed command. This sed command is needed because there are HTML tags in the results returned by grep. These will be interpreted by your browser, displaying a mess.
   To just print the HTML tags without interpreting them, we need to pipe the results through a sed command that:
   - converts all < characters to `<`
   - converts all > characters to `>`
   - The command is tricky to write because "&" has special meaning to sed and must be escaped.

Some enhancements

- change the output so the user can actually click on the pages returned.
- Consider where there are spaces in the argument (multiple search words), etc.

Some further enhancements

- If you have more than 2 levels of web pages you may write them out explicitly as `*/*/*html` etc., or get a recursive grep, or use recursive find first to build the filespec:

```
cd /users/homes/me/public_html

filespec=`find . -type f -name "*html" | tr '\n' ' '`

grep -i "$argument" $filespec
```

  - Since each search will be using the same file list, it would be more efficient to pre-build the list once, and cache it in a file, and then:

```
read filespec < filelist.txt

grep -i "$argument" $filespec
```

- The pages are not ranked in order of relevance, but only in the order in which `grep` finds them.
- Not easy to solve.

My search engine started out like this

- My search engine started out as a few lines of Shell like the above (plus a C++ input pre-processor for Web input security).
- It has since been re-written in PHP, but there is still a grep at the core.
- Obviously a heavy-duty search engine would pre-index the files in advance, rather than grep-ing them on the spot. But a grep is perfectly fine for a site of less than, say, 5,000 pages.

See search engine lab