MA194 – Using Unix

Instructor: Dan Chaney (chaney@ms.uky.edu)
Office hours: Mon 2-3, 606-257-2150

Class #25 - Tuesday, April 16, 1996 – Week 13

1996 NCAA Men's Basketball Champion: UK Wildcats!

Community Colleges: Tues 9:00-9:50 & 10:00-10:50 PM EST
UK/Lexington: Wed & Fri 9:00 PM EST, TCI Channel 16

URL: `http://www.ms.uky.edu/~ma194`
Mailing list: ma194-class@ms.uky.edu
To join the mailing list, send a note to ma194@ms.uky.edu

Tonight's Topics

- Homework, missed tapes and the IRS
- A shell help system
- Making forms in HTML
- Converting a shell script to CGI

The story so far...

1. Server reads a form and sends it to the client (form.html)

2. User fills out form and client sends data back to server

3. Server passes user's data to a program (specified in form.html)

4. CGI program reads data in from *stdin*, parses the output

5. CGI programs produces output (if necessary) and sends back to server

A sample input to a CGI script

Here is the sample input (*stdin*) to a CGI script with the variables **Uname**, **email** and **comments**.

```
Uname=dan+chaney&email=chaney@ms.uky.edu&comments=blank
```

Notice that the structure is:

variable=*value*

and how each pair is separated by an '&' and every space is marked by a '+'

Parsing the CGI input stream

1. Break the input up at the '&' mark
2. Break the input up at the '=' mark
3. Convert the '+' back into spaces
4. Take care of non-printable characters

The ‘awk‘ trick revealed

The **awk** program is used to break the data from the server up. ‘Parsing the input stream’ using the **awk** command looks like this:

```
.... | awk -F\= '{ print $1 ; print $2 }'
```

This takes the line coming in, puts everything to the left of the ‘=’ sign on one line and everything to the right on a second line.