

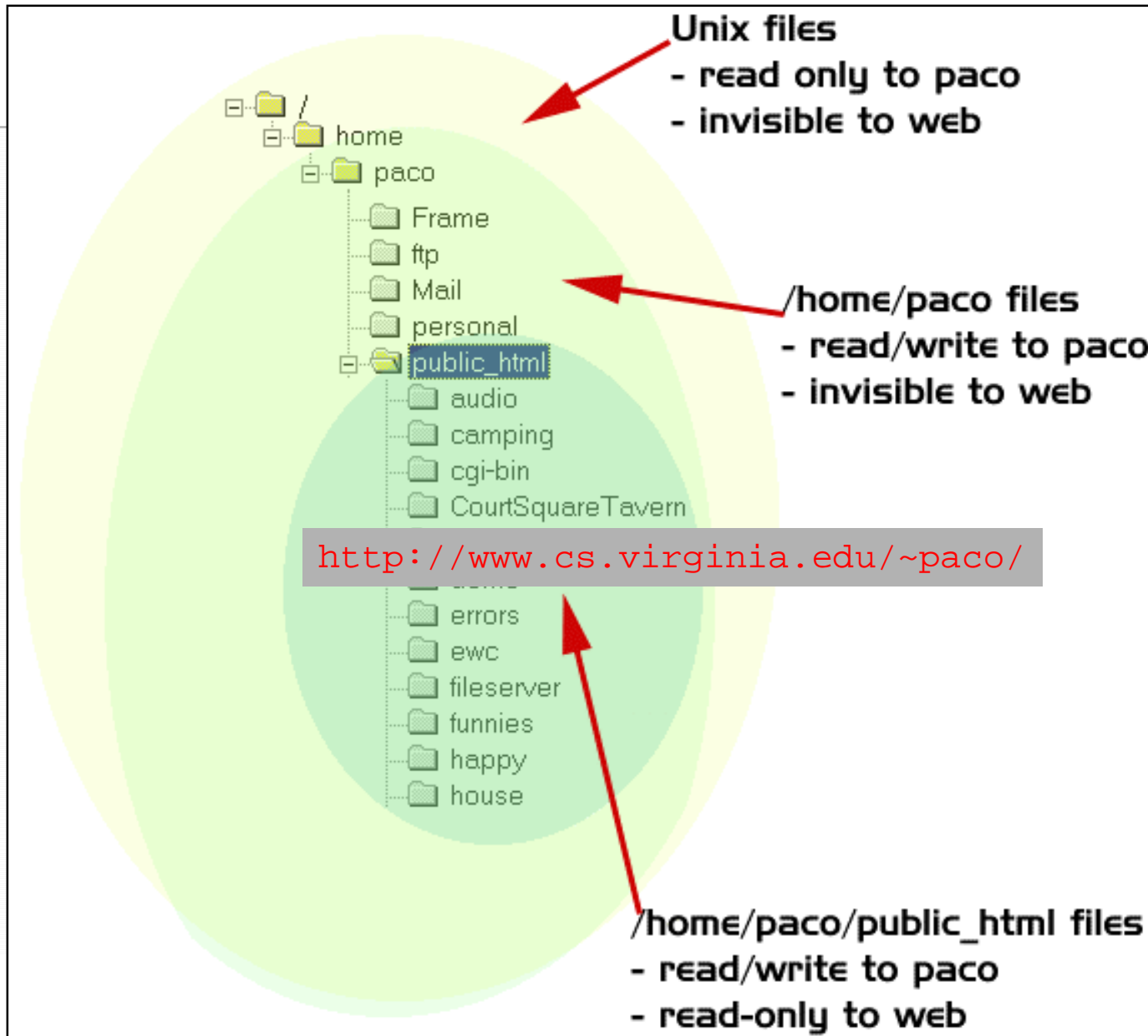
Server Side Includes  

Basic Dynamic Content with SSI
Paco Hope

This Lecture

- Brief **overview of unix** web server
- **Explain** server-side **mechanisms**
- Basic **syntax** of SSI
- Examples of **advanced** uses
- **References** to examples and documentation

The Web and Virtual Files



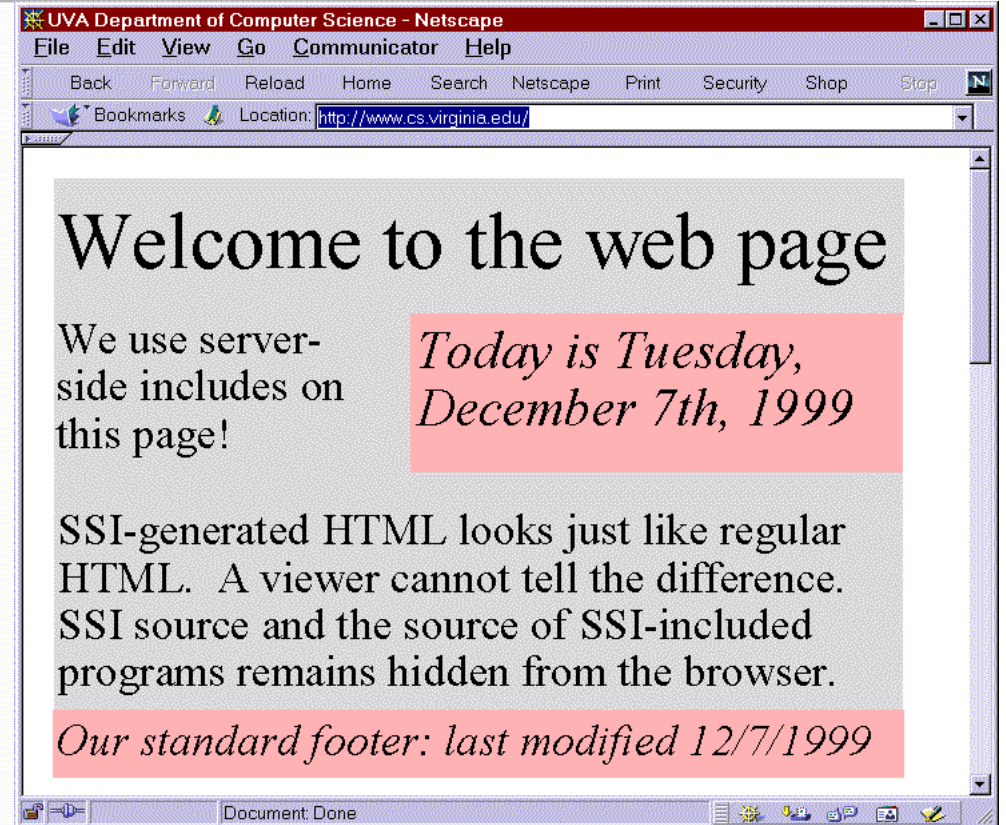
<http://www.cs.virginia.edu/~paco/>

Filesystems and URLs

- URLs point to files or directories
- Directory contents listed unless **index.html** (or other index) is present
- **/home/paco/public_html** (root of my web)
 - **http://www.cs.virginia.edu/~paco/**
- **/home/paco/public_html/test.html**
 - **http://www.cs.virginia.edu/~paco/test.html**
- **/home/paco/public_html/foo/index.html**
 - **http://www.cs.virginia.edu/~paco/foo/**

Server Side Mechanisms

- Server reads HTML file
- Server finds directives embedded in HTML
- Server executes directives



Step 1: Tag the File

- Make your file executable (UVA CS)
 - `chmod a+x file.html`
- Naming convention (other sites)
 - `file.shtml`
 - `file.html-ssi`
- If the file isn't tagged, the web server does not consider processing it for SSI

Step 2: Directives in HTML

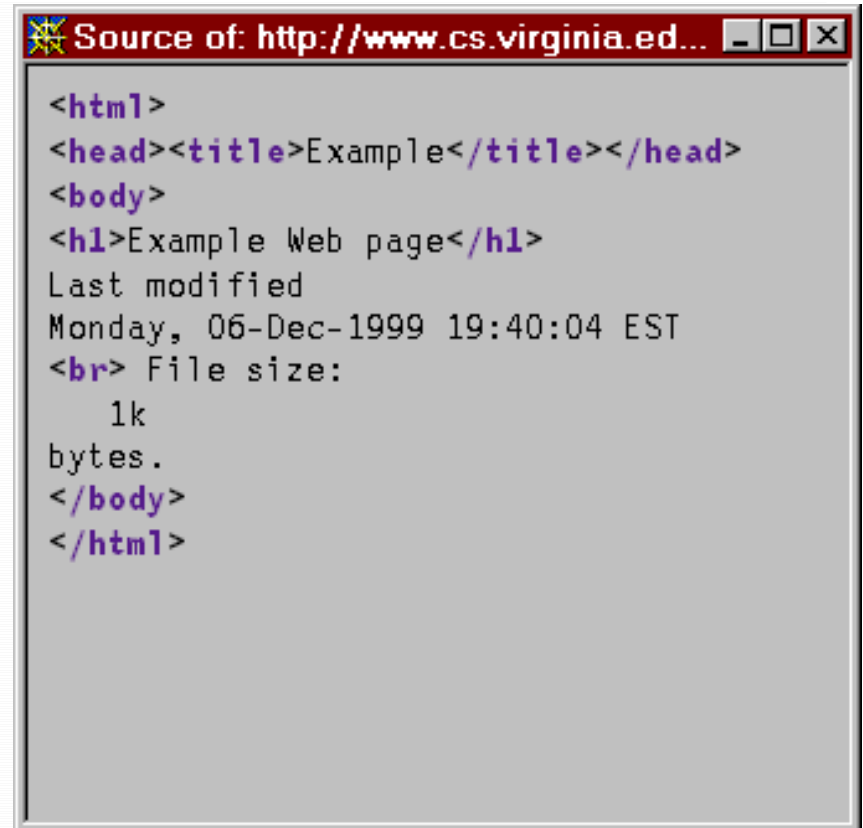
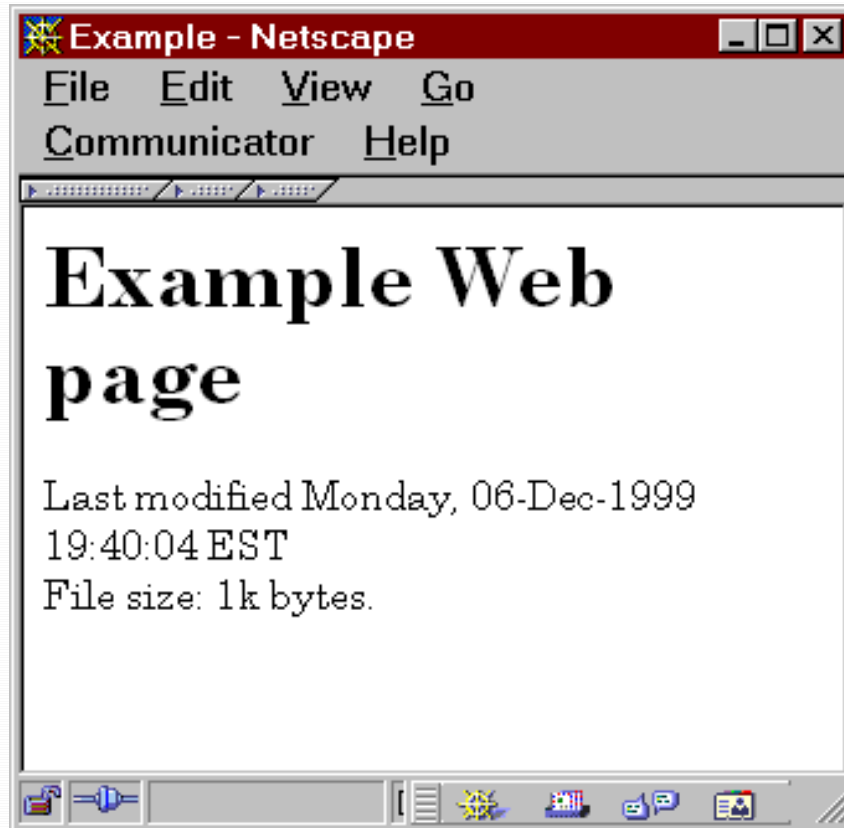
- Look like comments
- Follow common unix conventions
 - Environment
 - Expressions
 - [strftime\(\)](#)
- Don't appear in output if processed

```
<!--#include -->  
<!--#fsize -->  
<!--#if -->  
<!--#then -->  
<!--#else -->  
<!--#endif -->  
<!--#set -->
```

Example SSI and HTML

```
<html>
<head><title>Example</title></head>
<body>
<h1>Example Web page</h1>
Last modified
<!--#flastmod virtual="ex1.html" -->
<br> File size:
<!--#fsize      virtual="ex1.html" -->
bytes.
</body>
</html>
```

Resulting Output



Imperative Directives

`#config`

configure behavior

`#echo`

print variable values

`#exec`

CGI or commands

`#filesize`

print the size of a file

`#lastmod`

print a file's mod time

`#include`

include another file or CGI script in output

`#printenv`

print all environment variables (debugging)

`#set`

set an environment variable

Using Imperatives

- "virtual=..." refers to file in web space
- Virtual references are relative to current file
- "file=..." refers to absolute file in the file system
- "cmd=..." is absolute, too.

Examples

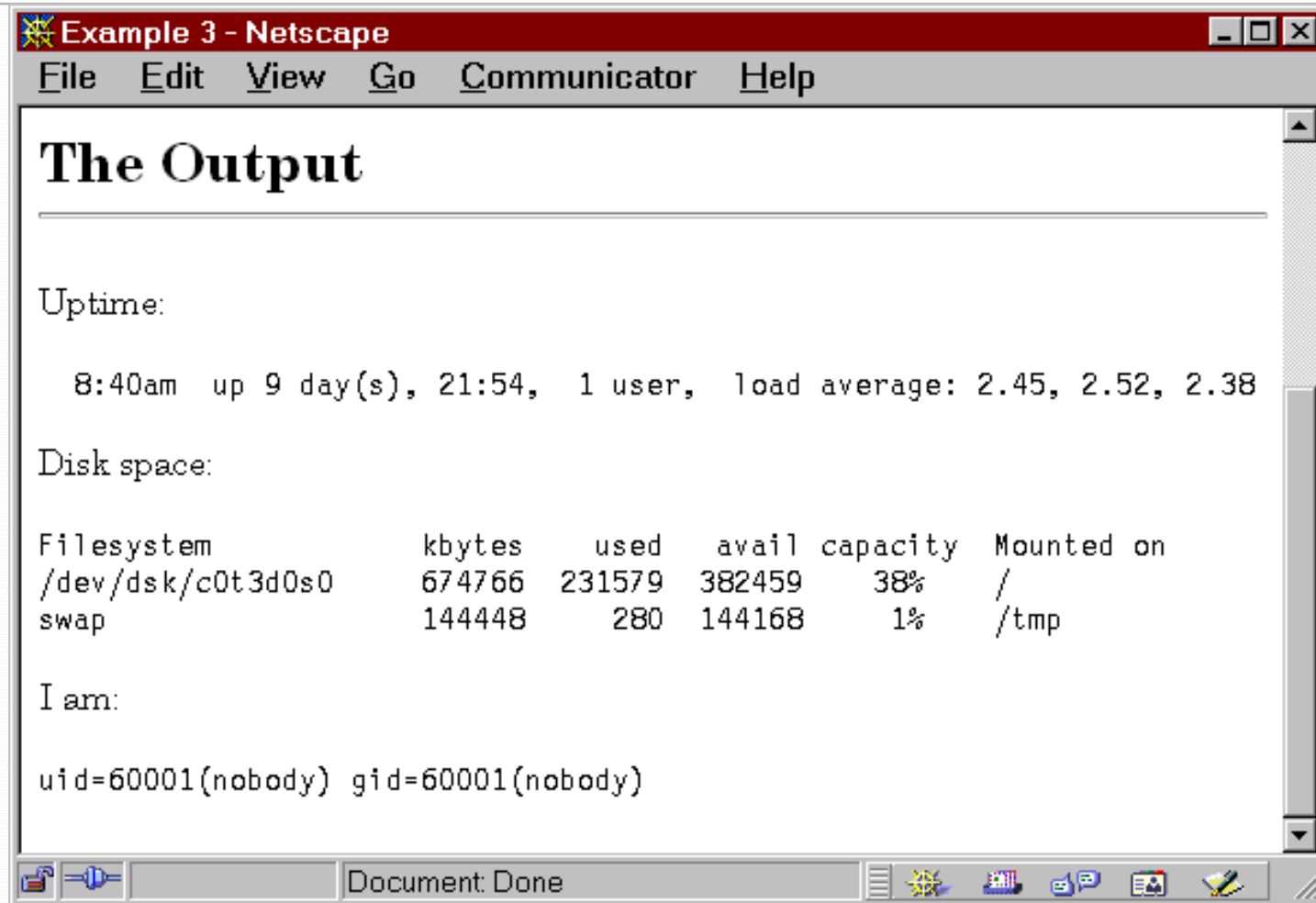
```
#include
    virtual="footer.html"
#include virtual="/cgi-
    bin/count.cgi"
#exec cgi="/cgi-
    bin/count.cgi"
#flastmod
    file="/home/paco/.plan"
#exec cmd="/bin/uptime"
```

Imperative Example

- Show how long the server has been up and how much disk space is free.
- Use standard unix commands
- Commands run as *nobody*

```
<html>
<head><title>Example 3
  </title></head><body>
<!--#exec
  cmd="/bin/uptime" -->
<!--#exec
  cmd="/bin/df -kl" -->
I am:
<!--#exec
  cmd="/bin/id" -->
</body>
</html>
```

Example Output



Control Flow Directives

Basic control directives:

#if

#then

#elif

#else

#endif

Expressions

➤ **Basic comparisons**

(=, !=, <, >, etc)

➤ **Combinations**

(a != b) && (b = c)

➤ **Regex Patterns**

`$var = '/^ftp.*/'`

Using Control Flow

Fun with time-based text

Depending on the
time of day, the
page offers
different
salutations

[Online Example](#)

If it's before 12:00pm,
say "good morning"

If it's between 12:00pm
and 6:00pm, say "good
afternoon"

If it's between 6:00pm
and midnight, say
"good evening"

Example Code

```
<!--#config timefmt="%H" --> (time var now prints only hour)
<!--#set var="hour" value="$DATE_LOCAL" -->
<!--#if expr="$hour < 12" -->
Good Morning
<!--#elif expr="($hour >= 12) && ($hour < 18)" -->
Good Afternoon
<!--#elif expr="$hour >= 18" -->
Good Evening
<!--#endif -->
```

<http://www.cs.virginia.edu/~paco/talks/web/lecture2/ex2.html>

Ridiculously Complex Example

- Create a shell command that generates random numbers.
- Convert them to hex.
- Assign them as background colors for a table or web page.
- (Should do this in PHP or CGI instead)

Random Color (Example 4)



Example 4

This command runs like a unix shell command, and gives us a two digit hex string representing a decimal value between 64 and 256.

```
/usr/cs/bin/bash -c 'echo "scale=2; obase=16; $RANDOM/32768*192+64" | bc | cut -d . -f 1 | tr -d \\n'
```

We call it 3 times to get an RGB color value like this: AB3E77. We then plug that value into the table as it's background color. This is crude and difficult, and it points out the utility of CGI.

<h2>Random color around this paragraph</h2>

Click reload to see a different color

The code

```
<!--#set var="randcolor" value="/usr/cs/bin/bash -c 'echo \"scale=2; obase=16; \\$RANDOM/32768*192+64\" | bc | bc | cut -d . -f 1 | tr -d \\n' -->  
<p>  
<table border="1" cellpadding="5" width="80%"  
  bgcolor="#<!--#exec cmd="$randcolor" --><!--#exec cmd="$randcolor" --><!--#exec cmd="$randcolor" -->">  
<tr><td align="center"><h1>Random color around this paragraph</h1></td></tr>  
</table>
```

Document: Done

Fun Pseudo-Randomness

- Get a bunch of quotations
- Use the last digit of the current time (in seconds) to select which quote to show
- Seems to change often, but it depends on when the person visits
- Example of regex matching

The Code

```
<!--#config timefmt="%S" -->
<!--#set      var="sec" value="$DATE_LOCAL" -->
<!--#if      expr="$sec = /[13579]/" -->
How do I set my Laser printer to "Stun"?
<!--#else -->
Me hav'em heap trouble. - Tonto the programmer
<!--#endif -->
```

- If the seconds are odd, show one quote
- If the seconds are even, show the other

<http://www.cs.virginia.edu/~paco/talks/web/lecture2/ex5.html>

Last Cool Example: Errors

- Apache allows **error documents** for various errors
- Configurable per-server, per-user, or **per-directory**
- Excellent use of SSI

Steps to Error Docs

- Create **.htaccess** file
- Put these kinds of lines in it:

```
ErrorDocument 404  
    /~bah6f/errors/general404.html
```
- Create the error doc web pages

Example 404 Document

```
<font color="Brown"><pre>
Error:    404 Not Found
Time:     <!--#echo var="DATE_LOCAL" -->
Browser:  <!--#echo var="HTTP_USER_AGENT" -->
From:     <!--#echo var="REMOTE_HOST" -->
          (<!--#echo var="REMOTE_ADDR" -->)
Document: <!--#echo var="REQUEST_URI" -->
<!--#if expr="$HTTP_REFERER" -->Referring Page:
         <!--#echo var="HTTP_REFERER" -->
<!--#endif -->
</pre></font>
```