

Dreadhawk's Fingering Utility

Who would you like to finger today?

Username?

Fingered user = 'gzimmer'

Login: gzimmer Name: Guy Zimmerman
Directory: /home/classes/gzimmer Shell: /bin/bash
Last login Wed Sep 10 23:00 (EDT) on pts/1 from cpe-24-210-133-213.woh.res.rr.com
No mail.
Project:
This space for rent.
Plan:
This space for rent.

DONE

voyager

apache

Wed Dec 10 20:02:19 EST 2014

total 92

```
drwxr-s---.  5 webgroup 4096 Dec 10 19:43 .
drwxrwsr-x. 83 webgroup 4096 Nov 25 10:06 ..
-rw-----.  1 webgroup  775 Dec  9 18:21 .bash_history
-rw-r-sr--.  1 users      18 Sep 25 22:13 .bash_logout
-rw-r-sr--.  1 users     176 Sep 25 22:13 .bash_profile
-rw-r-sr--.  1 users     134 Nov 22  2013 .bashrc
-rw-r-sr--.  1 users     500 May  7  2013 .emacs
drwxr-sr-x.  3 users     4096 Dec  9 11:57 .gnome2
drwxr-sr-x.  4 users     4096 Oct  7  2013 .mozilla
-rw-r-----.  1 webgroup  216 Dec  8 21:53 0helloWorld.perl
-rw-r-----.  1 webgroup  308 Dec  8 21:53 1commandLineArg.perl
-rw-r-----.  1 webgroup  624 Dec  8 21:53 2areTooFewArg.perl
-rw-r-----.  1 webgroup  424 Dec  8 21:53 3unixCommand.perl
-rw-r-----.  1 webgroup   0 Dec  9 12:17 3unixCommandOutputFile.txt
drwxr-s---.  2 webgroup 4096 Dec  9 16:05 cs3270Sample
-rwxr-x--x.  1 webgroup 2202 Dec 10 19:58 finger.cgi
-rwxr-x--x.  1 webgroup  441 Dec 10 20:00 finger.html
-rw-r-----.  1 webgroup   33 Dec  3 08:23 instScript
-rwxr-x--x.  1 webgroup 1571 Dec 10 19:42 sampleCGIdocument.cgi
-rw-r-----.  1 webgroup 1130 Dec  8 21:53 sampleCGIdocument.cgi.txt
```

```
-rwxr-x--x.  1 webgroup  322 Dec 10 18:15 sampleHtmldocument.html
-rw-r-----.  1 webgroup  242 Dec  9 18:08 test.perl
-rw-r-----.  1 webgroup 6324 Dec  8 21:53 typescript
```

HERE IS THE GCI!

```
#!/usr/bin/perl
#Addison Snyder
MAIN:
{
```

```
#print &PrintHeader;
&ReadParse;
```

```
    $NAME = $in{'myName'};
    $OUTPUT = `finger $NAME`;
    $OUTPUTEDIT = qq{<xmp>$OUTPUT
```

```
#Retrieves and prepares finger command string
```

```
}; $CGI = `cat finger.cgi`; $CGIEDIT = qq{
```

```
$CGI
```

```
}; #Retrieves and preps cgi file output #
```

```
    $HTML = `cat finger.html`;
    $HTMLEEDIT = qq{<xmp>$HTML
```

```
}; #Retrieves and preps html file output #
```

```
    local $/ = undef;
    open FILE, "instScript" or die "Couldn't open file: $!";
    binmode FILE;
    $INSTSCRIPT = <FILE>;
    close FILE;
```

```
#Opens instScript file and gets string of commands
```

```
    $COMMAND = `$INSTSCRIPT`;
    $COMMANDEEDIT = qq{<xmp>$COMMAND
```

}; #issues the instScript commands and retrieves/preps output #

```
    &echo();
}
```

```
sub echo{                                     #Echo function for displaying results

print "Content-type: text/html\n\n <html> <head> <title>";      #makes sure that output is predominantly html
print "</title></head>\n";                                         #head stores rudimentary info about page (i.e. title)
print "                                                           #This section of html prints all of the prepped materials
"
<BODY>
<H1>Fingered user = '$NAME'</H1>
<H5>$OUTPUTEDIT<H5>
<H1>DONE<H1>
<H3>$COMMANDEDIT<H3>
<HL>
<H3>HERE IS THE GCI!<H3>
<H5>$CGIEDIT<H5>
<H3>HERE IS THE HTML!<H3>
<H5>$HTMLEEDIT<H5>
</BODY>
</HTML>
";
}
```

```
#      Got this routine off of web.
#      Do not remember the source.
```

```
sub ReadParse {                               #This looks complicated and messy... NOT gonna touch it...
# slightly modified from the usual readparse
local (*in) = @_ if @_; local ($i, $loc, $key, $val);
if ($ENV{'REQUEST_METHOD'} eq "GET")
{ $in = $ENV{'QUERY_STRING'}; }
elsif ($ENV{'REQUEST_METHOD'} eq "POST")
{ read(STDIN,$in,$ENV{'CONTENT_LENGTH'}); }
@in = split(/&/,$in);
foreach $i (0 .. $#in) {
    $in[$i] =~ s/\+ / /g; ($key, $val) = split(/=/, $in[$i], 2);
    $FOUND=0; foreach (@FIELDS) { $FOUND=1 if ($key eq $_); }
    if (($key ne 'TO') && ($key ne 'SUBJECT') && ($key ne 'REPLY-TO') && ($FOUND==0))
    { push(@FIELDS,$key); }
    $key =~ s/%(..)/pack("c",hex($1))/ge;
    $val =~ s/%(..)/pack("c",hex($1))/ge;
}
```

```
$in{$key} .= "\0" if (defined($in{$key}));  
$in{$key} .= $val; }  
  
return 1;  
}
```

HERE IS THE HTML!

```
<HTML>  
<HEAD>  
</HEAD>  
  
<H1>Dreadhawk's Fingering Utility </H1>      <!-- Name is trademarked.. -->  
<h3>Who would you like to finger today?</h3> <!-- So is this -->  
<HR>  
<FORM METHOD="POST" ACTION="finger.cgi">      <!-- redirects to cgi for processing -->  
<DL>  
<DD>Username?  <INPUT TYPE="text" NAME="myName" SIZE=15 VALUE=""><BR>  <!-- Form -->  
</DL>  
<p>  
<INPUT TYPE="submit" VALUE="Submit User?">   <!-- Submit button -->  
</FORM>  
<HR>  
</BODY>  
</HTML>
```

Dreadhawk's Fingering Utility

Who would you like to finger today?

Username?

Fingered user = 'kresman'

Login: kresman Name: Ray Kresman
Directory: /home/classes/kresman Shell: /bin/bash
Last login Mon Dec 8 23:01 (EST) on pts/13 from 50.51.61.77
No mail.
No Plan.

DONE

voyager
apache

Wed Dec 10 20:02:34 EST 2014

total 92

```
drwxr-s---.  5 webgroup 4096 Dec 10 19:43 .
drwxrwsr-x. 83 webgroup 4096 Nov 25 10:06 ..
-rw-----.  1 webgroup  775 Dec  9 18:21 .bash_history
-rw-r-sr--.  1 users    18 Sep 25 22:13 .bash_logout
-rw-r-sr--.  1 users    176 Sep 25 22:13 .bash_profile
-rw-r-sr--.  1 users    134 Nov 22  2013 .bashrc
-rw-r-sr--.  1 users    500 May  7  2013 .emacs
drwxr-sr-x.  3 users    4096 Dec  9 11:57 .gnome2
drwxr-sr-x.  4 users    4096 Oct  7  2013 .mozilla
-rw-r-----.  1 webgroup  216 Dec  8 21:53 0helloWorld.perl
-rw-r-----.  1 webgroup  308 Dec  8 21:53 1commandLineArg.perl
-rw-r-----.  1 webgroup  624 Dec  8 21:53 2areTooFewArg.perl
-rw-r-----.  1 webgroup  424 Dec  8 21:53 3unixCommand.perl
-rw-r-----.  1 webgroup   0 Dec  9 12:17 3unixCommandOutputFile.txt
drwxr-s---.  2 webgroup 4096 Dec  9 16:05 cs3270Sample
-rwxr-x--x.  1 webgroup 2202 Dec 10 19:58 finger.cgi
-rwxr-x--x.  1 webgroup  441 Dec 10 20:00 finger.html
-rw-r-----.  1 webgroup  33 Dec  3 08:23 instScript
-rwxr-x--x.  1 webgroup 1571 Dec 10 19:42 sampleCGIdocument.cgi
-rw-r-----.  1 webgroup 1130 Dec  8 21:53 sampleCGIdocument.cgi.txt
-rwxr-x--x.  1 webgroup  322 Dec 10 18:15 sampleHtmldocument.html
-rw-r-----.  1 webgroup  242 Dec  9 18:08 test.perl
```

-rw-r-----. 1 webgroup 6324 Dec 8 21:53 typescript

HERE IS THE GCI!

```
#!/usr/bin/perl
#Addison Snyder
MAIN:
{

#print &PrintHeader;
&ReadParse;

    $NAME = $in{'myName'};
    $OUTPUT = `finger $NAME`;
    $OUTPUTEDIT = qq{<xmp>$OUTPUT
#Retrieves and prepares finger command string

}; $CGI = `cat finger.cgi`; $CGIEDIT = qq{

$CGI

}; #Retrieves and preps cgi file output #

    $HTML = `cat finger.html`;
    $HTMLEEDIT = qq{<xmp>$HTML

}; #Retrieves and preps html file output #

    local $/ = undef;
    open FILE, "instScript" or die "Couldn't open file: $!";
    binmode FILE;
    $INSTSCRIPT = <FILE>;
    close FILE;
#Opens instScript file and gets string of commands

    $COMMAND = `$INSTSCRIPT`;
    $COMMANDEEDIT = qq{<xmp>$COMMAND

}; #issues the instScript commands and retrieves/preps output #
```



```

    &echo();
}

sub echo{                                     #Echo function for displaying results

print "Content-type: text/html\n\n <html> <head> <title>";      #makes sure that output is predominantly html
print "</title></head>\n";                                       #head stores rudimentary info about page (i.e. title)
print "                                                         #This section of html prints all of the prepped materials
"
<BODY>
<H1>Fingered user = '$NAME'</H1>
<H5>$OUTPUTEDIT<H5>
<H1>DONE<H1>
<H3>$COMMANDEDIT<H3>
<HL>
<H3>HERE IS THE GCI!<H3>
<H5>$CGIEDIT<H5>
<H3>HERE IS THE HTML!<H3>
<H5>$HTMLEEDIT<H5>
</BODY>
</HTML>
";
}

#      Got this routine off of web.
#      Do not remember the source.

sub ReadParse {                               #This looks complicated and messy... NOT gonna touch it...
# slightly modified from the usual readparse
local (*in) = @_ if @_; local ($i, $loc, $key, $val);
if ($ENV{'REQUEST_METHOD'} eq "GET")
{ $in = $ENV{'QUERY_STRING'}; }
elsif ($ENV{'REQUEST_METHOD'} eq "POST")
{ read(STDIN,$in,$ENV{'CONTENT_LENGTH'}); }
@in = split(/&/,$in);
foreach $i (0 .. $#in) {
    $in[$i] =~ s/\+ / /g; ($key, $val) = split(/=/, $in[$i], 2);
    $FOUND=0; foreach (@FIELDS) { $FOUND=1 if ($key eq $_); }
    if (($key ne 'TO')&&($key ne 'SUBJECT')&&($key ne 'REPLY-TO')&&($FOUND==0))
    { push(@FIELDS,$key); }
    $key =~ s/%(..)/pack("c",hex($1))/ge;
    $val =~ s/%(..)/pack("c",hex($1))/ge;
    $in{$key} .= "\0" if (defined($in{$key}));
    $in{$key} .= $val; }
}

```

```
    return 1;
}
```

HERE IS THE HTML!

```
<HTML>
<HEAD>
</HEAD>

<H1>Dreadhawk's Fingering Utility </H1>      <!-- Name is trademarked.. -->
<h3>Who would you like to finger today?</h3>  <!-- So is this -->
<HR>
<FORM METHOD="POST" ACTION="finger.cgi">      <!-- redirects to cgi for processing -->
<DL>
<DD>Username?  <INPUT TYPE="text" NAME="myName" SIZE=15 VALUE=""><BR>  <!-- Form -->
</DL>
<p>
<INPUT TYPE="submit" VALUE="Submit User?">    <!-- Submit button -->
</FORM>
<HR>
</BODY>
</HTML>
```

Fingered user = 'kresman gzimmer'

Login: kresman Name: Ray Kresman
Directory: /home/classes/kresman Shell: /bin/bash
Last login Mon Dec 8 23:01 (EST) on pts/13 from 50.51.61.77
No mail.
No Plan.

Login: gzimmer Name: Guy Zimmerman
Directory: /home/classes/gzimmer Shell: /bin/bash
Last login Wed Sep 10 23:00 (EDT) on pts/1 from cpe-24-210-133-213.woh.res.rr.com
No mail.
Project:
This space for rent.
Plan:
This space for rent.

DONE

voyager

apache

Wed Dec 10 20:02:55 EST 2014

total 92

```
drwxr-s---.  5 webgroup 4096 Dec 10 19:43 .
drwxrwsr-x. 83 webgroup 4096 Nov 25 10:06 ..
-rw-----.  1 webgroup  775 Dec  9 18:21 .bash_history
-rw-r-sr--.  1 users      18 Sep 25 22:13 .bash_logout
-rw-r-sr--.  1 users     176 Sep 25 22:13 .bash_profile
-rw-r-sr--.  1 users     134 Nov 22  2013 .bashrc
-rw-r-sr--.  1 users     500 May  7  2013 .emacs
drwxr-sr-x.  3 users     4096 Dec  9 11:57 .gnome2
drwxr-sr-x.  4 users     4096 Oct  7  2013 .mozilla
-rw-r-----.  1 webgroup  216 Dec  8 21:53 0helloWorld.perl
-rw-r-----.  1 webgroup  308 Dec  8 21:53 1commandLineArg.perl
-rw-r-----.  1 webgroup  624 Dec  8 21:53 2areTooFewArg.perl
-rw-r-----.  1 webgroup  424 Dec  8 21:53 3unixCommand.perl
-rw-r-----.  1 webgroup   0 Dec  9 12:17 3unixCommandOutputFile.txt
drwxr-s---.  2 webgroup 4096 Dec  9 16:05 cs3270Sample
-rwxr-x--x.  1 webgroup 2202 Dec 10 19:58 finger.cgi
```

```
-rwxr-x--x.  1 webgroup  441 Dec 10 20:00 finger.html
-rw-r-----.  1 webgroup   33 Dec  3 08:23 instScript
-rwxr-x--x.  1 webgroup 1571 Dec 10 19:42 sampleCGIdocument.cgi
-rw-r-----.  1 webgroup 1130 Dec  8 21:53 sampleCGIdocument.cgi.txt
-rwxr-x--x.  1 webgroup  322 Dec 10 18:15 sampleHtmldocument.html
-rw-r-----.  1 webgroup  242 Dec  9 18:08 test.perl
-rw-r-----.  1 webgroup 6324 Dec  8 21:53 typescript
```

HERE IS THE GCI!

```
#!/usr/bin/perl
#Addison Snyder
MAIN:
{
```

```
#print &PrintHeader;
&ReadParse;
```

```
    $NAME = $in{'myName'};
    $OUTPUT = `finger $NAME`;
    $OUTPUTEDIT = qq{<xmp>$OUTPUT
```

#Retrieves and prepares finger command string

```
}; $CGI = `cat finger.cgi`; $CGIEDIT = qq{
```

```
$CGI
```

```
}; #Retrieves and preps cgi file output #
```

```
    $HTML = `cat finger.html`;
    $HTMLEEDIT = qq{<xmp>$HTML
```

```
}; #Retrieves and preps html file output #
```

```
    local $/ = undef;
    open FILE, "instScript" or die "Couldn't open file: $!";
    binmode FILE;
    $INSTSCRIPT = <FILE>;
    close FILE;
```

#Opens instScript file and gets string of commands

```

$COMMAND = ` $INSTSCRIPT `;
$COMMANDEDIT = qq{<xmp>$COMMAND

}; #issues the instScript commands and retrieves/preps output #

    &echo();
}

sub echo{                                     #Echo function for displaying results

print "Content-type: text/html\n\n <html> <head> <title>";      #makes sure that output is predominantly html
print "</title></head>\n";                                         #head stores rudimentary info about page (i.e. title)
print "                                                           #This section of html prints all of the prepped materials
"
<BODY>
<H1>Fingered user = '$NAME'</H1>
<H5>$OUTPUTEDIT<H5>
<H1>DONE<H1>
<H3>$COMMANDEDIT<H3>
<HL>
<H3>HERE IS THE GCI!<H3>
<H5>$CGIEDIT<H5>
<H3>HERE IS THE HTML!<H3>
<H5>$HTMLEEDIT<H5>
</BODY>
</HTML>
";
}

#      Got this routine off of web.
#      Do not remember the source.

sub ReadParse {                               #This looks complicated and messy... NOT gonna touch it...
# slightly modified from the usual readparse
local (*in) = @_ if @_; local ($i, $loc, $key, $val);
if ($ENV{'REQUEST_METHOD'} eq "GET")
    { $in = $ENV{'QUERY_STRING'}; }
elsif ($ENV{'REQUEST_METHOD'} eq "POST")
    { read(STDIN,$in,$ENV{'CONTENT_LENGTH'}); }
@in = split(/&/,$in);
foreach $i (0 .. $#in) {

```

```

$in[$i] =~ s/\+ /g;    ($key, $val) = split(/=/,$in[$i],2);
$FOUND=0;    foreach (@FIELDS) { $FOUND=1 if ($key eq $_); }
if (($key ne 'TO')&&($key ne 'SUBJECT')&&($key ne 'REPLY-TO')&&($FOUND==0))
    { push(@FIELDS,$key); }
$key =~ s/%(..)/pack("c",hex($1))/ge;
$val =~ s/%(..)/pack("c",hex($1))/ge;
$in{$key} .= "\0" if (defined($in{$key}));
$in{$key} .= $val; }

return 1;
}

```

HERE IS THE HTML!

```

<HTML>
<HEAD>
</HEAD>

<H1>Dreadhawk's Fingering Utility </H1>      <!-- Name is trademarked.. -->
<h3>Who would you like to finger today?</h3>  <!-- So is this -->
<HR>
<FORM METHOD="POST" ACTION="finger.cgi">      <!-- redirects to cgi for processing -->
<DL>
<DD>Username?  <INPUT TYPE="text" NAME="myName" SIZE=15 VALUE=""><BR>  <!-- Form -->
</DL>
<p>
<INPUT TYPE="submit" VALUE="Submit User?">    <!-- Submit button -->
</FORM>
<HR>
</BODY>
</HTML>

```

Dreadhawk's Fingering Utility

Who would you like to finger today?

Username?
