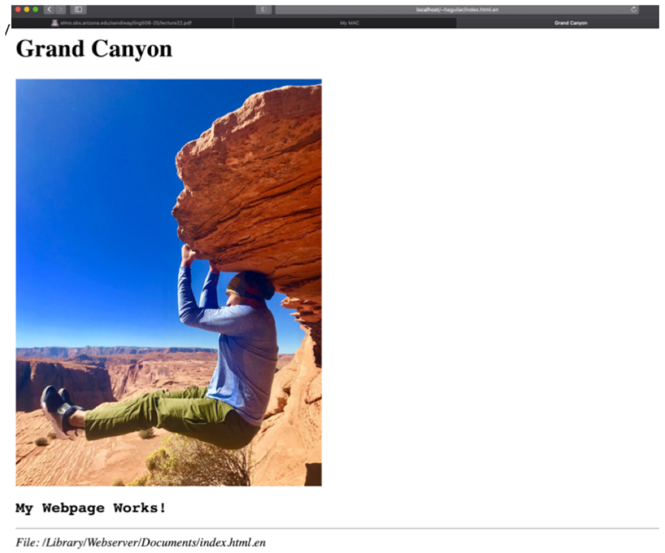
LING 408/508: Computational Techniques for Linguists

Lecture 23

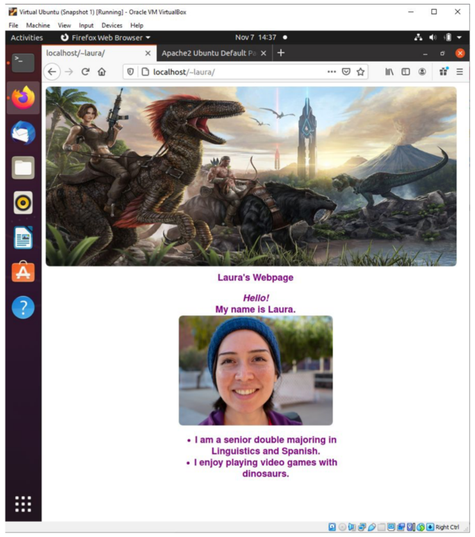
Today's Topics

- your own webserver (hw8:
  - did everyone manage to get two different pages going?
  - These are cool examples…


Grand Canyon

`My Webpage Works!`

File: /Library/Webserver/Documents/index.html.en



Laura's Webpage

Hello!
My name is Laura.

- **I am a senior double majoring in Linguistics and Spanish.**
- **I enjoy playing video games with dinosaurs.**

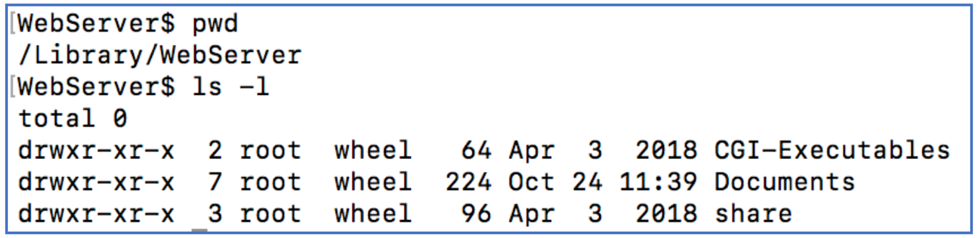
Today's Topics

- Today we will start running programs on the webserver itself.
  - in the default cgi-bin directory
  - inside your home directory
- Homework 9:
  - a bit like Homework 8
  - run cgi-bin scripts in both directories.
  - send screenshots

Apache2 Webserver on MacOS

Storage locations:

html pages:
`/Library/WebServer/Documents/`

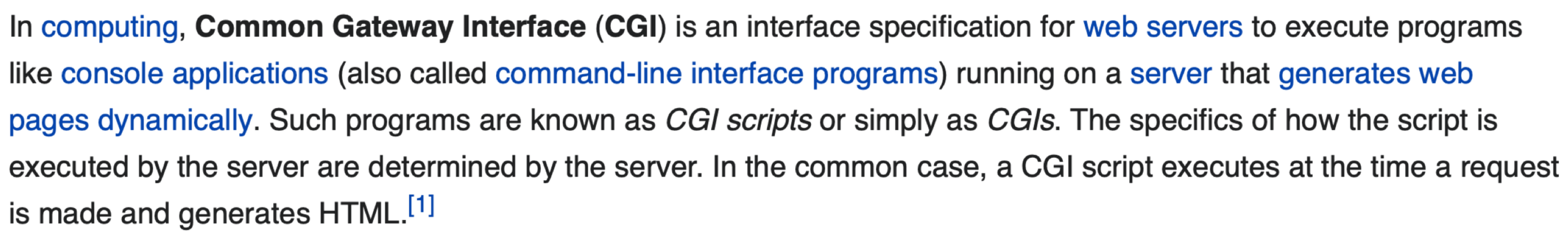


```
WebServer$ pwd
/Library/WebServer
WebServer$ ls -l
total 0
drwxr-xr-x  2 root  wheel   64 Apr  3  2018 CGI-Executables
drwxr-xr-x  7 root  wheel  224 Oct 24 11:39 Documents
drwxr-xr-x  3 root  wheel   96 Apr  3  2018 share
```

- cgi-bin directory:
  - **`/Library/WebServer/CGI-Executables/`**
- usage:
  - http://localhost/cgi-bin/test.cgi
    - permissions for *.cgi should be readable and executable
      - ls -l /Library/WebServer/CGI-Executables/
      - -rwxr-xr-x 1 root wheel   161 Oct 16 2014 get.cgi
      - -rwxr-xr-x 1 root wheel   125 Oct 21 2014 post.cgi
      - -rw-r--r-- 1 root wheel   113 Oct 27 16:06 test.cgi
  - to change permissions
    - **`sudo chmod 755 test.cgi`**

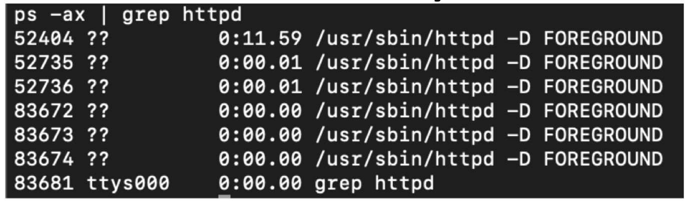
CGI-bin

- https://en.wikipedia.org/wiki/Common_Gateway_Interface


In computing, **Common Gateway Interface** (**CGI**) is an interface specification for web servers to execute programs like console applications (also called command-line interface programs) running on a server that generates web pages dynamically. Such programs are known as *CGI scripts* or simply as *CGIs*. The specifics of how the script is executed by the server are determined by the server. In the common case, a CGI script executes at the time a request is made and generates HTML.[1]

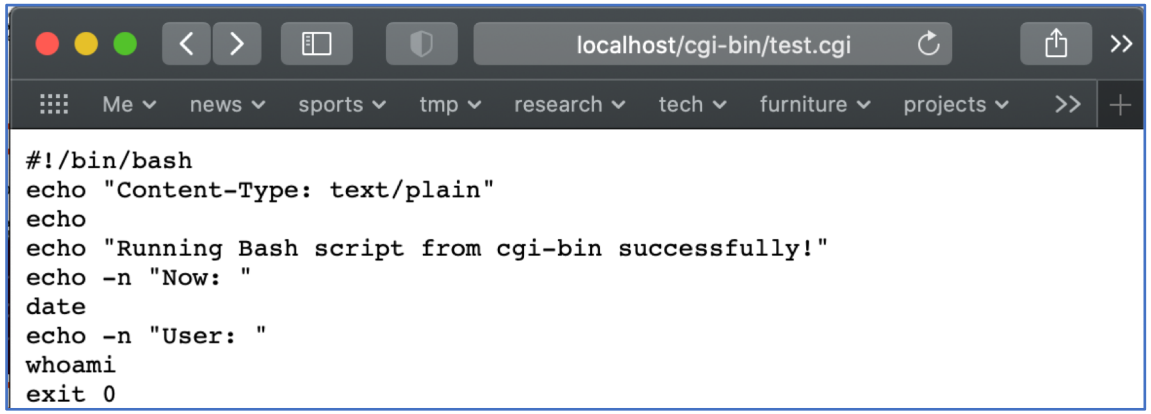
Apache2 Webserver on MacOS

- First, make sure your webserver is running:



```
ps -ax | grep httpd
52404 ??         0:11.59 /usr/sbin/httpd -D FOREGROUND
52735 ??         0:00.01 /usr/sbin/httpd -D FOREGROUND
52736 ??         0:00.01 /usr/sbin/httpd -D FOREGROUND
83672 ??         0:00.00 /usr/sbin/httpd -D FOREGROUND
83673 ??         0:00.00 /usr/sbin/httpd -D FOREGROUND
83674 ??         0:00.00 /usr/sbin/httpd -D FOREGROUND
83681 ttys000    0:00.00 grep httpd
```

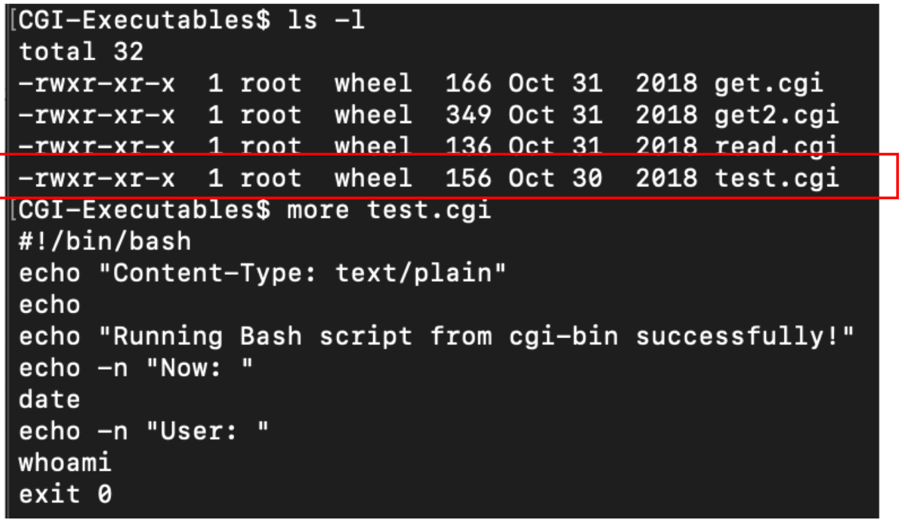
- On MacOS, here's a bash script (called `test.cgi`) to place in `/Library/WebServer/CGI-Executables`. When we try it, this might happen:



```
#!/bin/bash
echo "Content-Type: text/plain"
echo
echo "Running Bash script from cgi-bin successfully!"
echo -n "Now: "
date
echo -n "User: "
whoami
exit 0
```

Notice it prints the contents of the script instead of running it!

Apache2 Webserver on MacOS

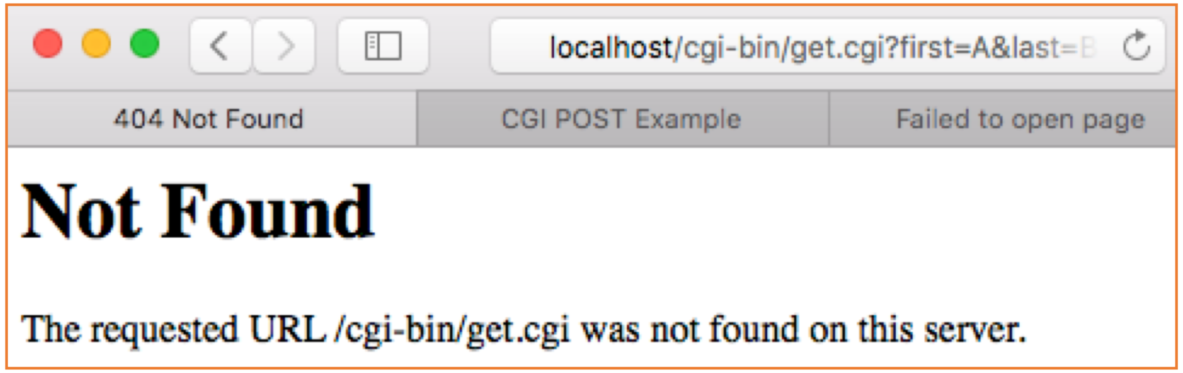


```
CGI-Executables$ ls -l
total 32
-rwxr-xr-x  1 root  wheel  166 Oct 31  2018 get.cgi
-rwxr-xr-x  1 root  wheel  349 Oct 31  2018 get2.cgi
-rwxr-xr-x  1 root  wheel  136 Oct 31  2018 read.cgi
-rwxr-xr-x  1 root  wheel  156 Oct 30  2018 test.cgi
CGI-Executables$ more test.cgi
#!/bin/bash
echo "Content-Type: text/plain"
echo
echo "Running Bash script from cgi-bin successfully!"
echo -n "Now: "
date
echo -n "User: "
whoami
exit 0
```

- This directory is owned by `root`.
- So you must create it using:
  - `sudo nano /Library/WebServer/CGI-Executables/test.cgi`

Apache2 Webserver on MacOS

- Assuming the webserver is running, possible symptoms:
  1. *prints some_program.cgi as plain text instead of executing it*
  2. *404 Not Found: some_program.cgi doesn't exist!*

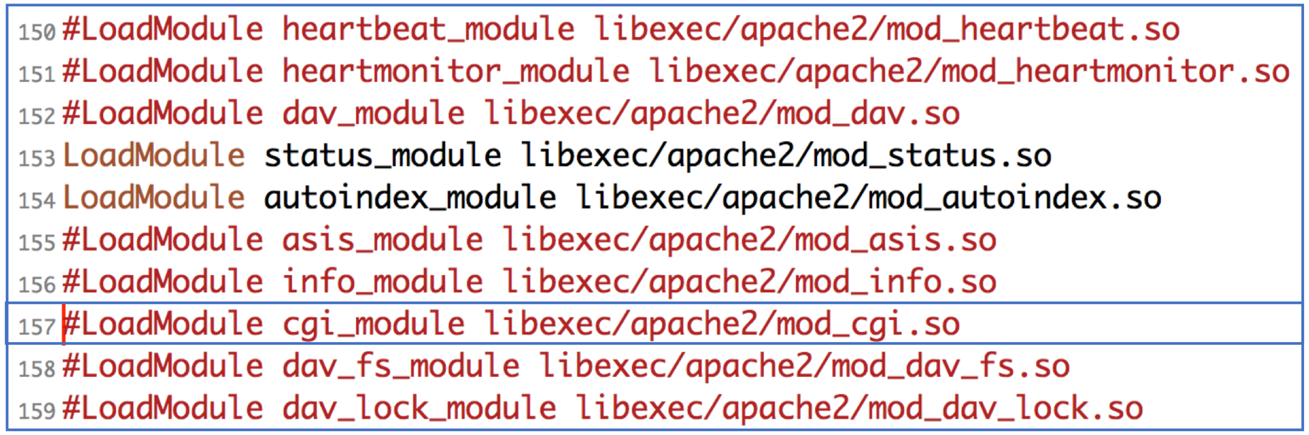


Not Found

The requested URL /cgi-bin/get.cgi was not found on this server.

Apache2 Webserver on MacOS

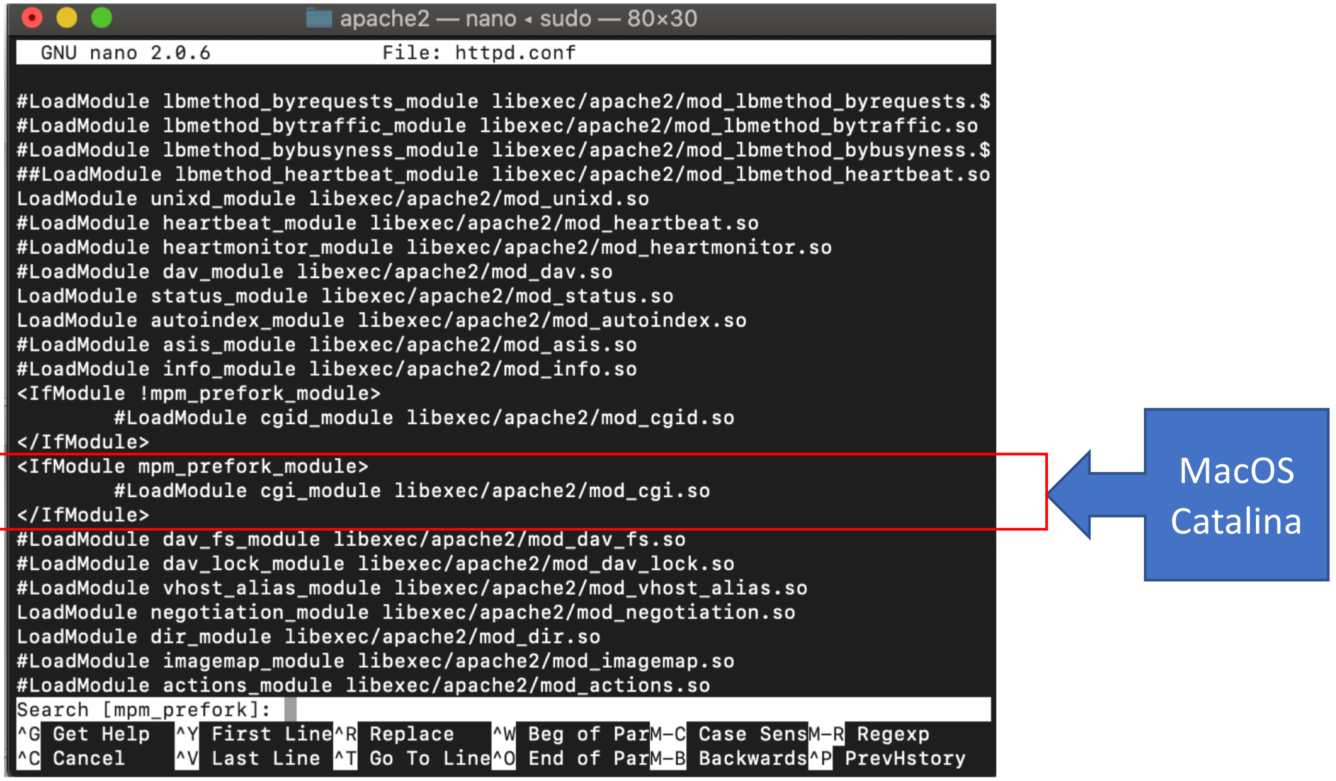
- By default, Apple ships Apache2 with the cgi module turned off.
- Enabling cgi-bin:
  1. uncomment **`cgi_module`** line in file **`/etc/apache2/httpd.conf`**
     - **`sudo nano /etc/apache2/httpd.conf`**
  2. restart apache2: **`sudo apachectl –k restart`**



```
#LoadModule heartbeat_module libexec/apache2/mod_heartbeat.so
#LoadModule heartmonitor_module libexec/apache2/mod_heartmonitor.so
#LoadModule dav_module libexec/apache2/mod_dav.so
LoadModule status_module libexec/apache2/mod_status.so
LoadModule autoindex_module libexec/apache2/mod_autoindex.so
#LoadModule asis_module libexec/apache2/mod_asis.so
#LoadModule info_module libexec/apache2/mod_info.so
#LoadModule cgi_module libexec/apache2/mod_cgi.so
#LoadModule dav_fs_module libexec/apache2/mod_dav_fs.so
#LoadModule dav_lock_module libexec/apache2/mod_dav_lock.so
```

MacOS Sierra

Apache2 Webserver on MacOS



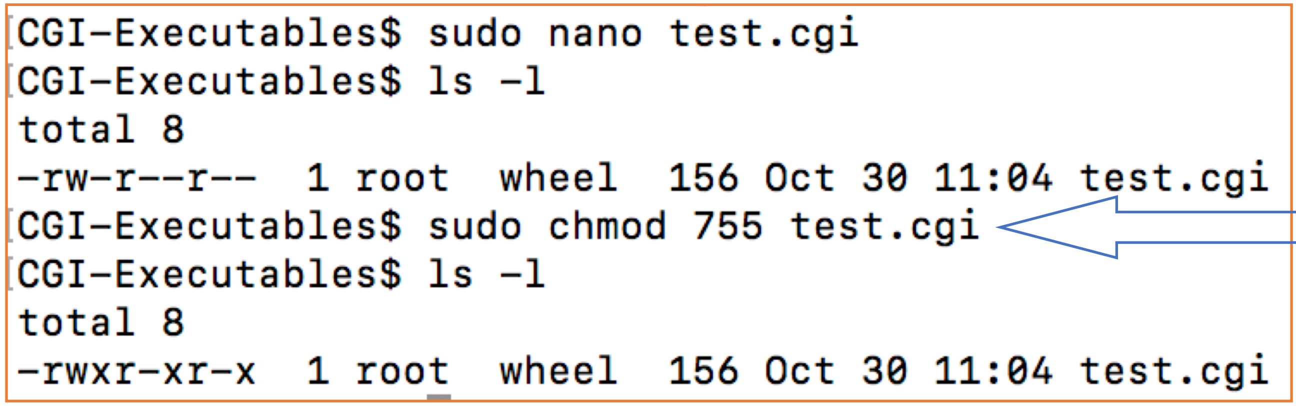
```
  GNU nano 2.0.6                 File: httpd.conf

#LoadModule lbmethod_byrequests_module libexec/apache2/mod_lbmethod_byrequests.$
#LoadModule lbmethod_bytraffic_module libexec/apache2/mod_lbmethod_bytraffic.so
#LoadModule lbmethod_bybusyness_module libexec/apache2/mod_lbmethod_bybusyness.$
##LoadModule lbmethod_heartbeat_module libexec/apache2/mod_lbmethod_heartbeat.so
LoadModule unixd_module libexec/apache2/mod_unixd.so
#LoadModule heartbeat_module libexec/apache2/mod_heartbeat.so
#LoadModule heartmonitor_module libexec/apache2/mod_heartmonitor.so
#LoadModule dav_module libexec/apache2/mod_dav.so
LoadModule status_module libexec/apache2/mod_status.so
LoadModule autoindex_module libexec/apache2/mod_autoindex.so
#LoadModule asis_module libexec/apache2/mod_asis.so
#LoadModule info_module libexec/apache2/mod_info.so
<IfModule !mpm_prefork_module>
        #LoadModule cgid_module libexec/apache2/mod_cgid.so
</IfModule>
<IfModule mpm_prefork_module>
        #LoadModule cgi_module libexec/apache2/mod_cgi.so
</IfModule>
#LoadModule dav_fs_module libexec/apache2/mod_dav_fs.so
#LoadModule dav_lock_module libexec/apache2/mod_dav_lock.so
#LoadModule vhost_alias_module libexec/apache2/mod_vhost_alias.so
LoadModule negotiation_module libexec/apache2/mod_negotiation.so
LoadModule dir_module libexec/apache2/mod_dir.so
#LoadModule imagemap_module libexec/apache2/mod_imagemap.so
#LoadModule actions_module libexec/apache2/mod_actions.so
Search [mpm_prefork]:
^G Get Help   ^Y First Line^R Replace   ^W Beg of ParM-C Case SensM-R Regexp
^C Cancel     ^V Last Line ^T Go To Line^O End of ParM-B Backwards^P PrevHstory
```

Apache2 Webserver on MacOS

- Simple bash script, let's call it **`test.cgi`**:

```
#!/bin/bash
echo "Content-Type: text/plain"
echo
echo "Running Bash script from cgi-bin successfully!"
echo -n "Now: "
date
echo -n "User: "
whoami
exit 0
```

Boilerplate (the browser expects):
`Content-Type: text/plain`
`<blank line>`



```
CGI-Executables$ sudo nano test.cgi
CGI-Executables$ ls -l
total 8
-rw-r--r--  1 root  wheel  156 Oct 30 11:04 test.cgi
CGI-Executables$ sudo chmod 755 test.cgi
CGI-Executables$ ls -l
total 8
-rwxr-xr-x  1 root  wheel  156 Oct 30 11:04 test.cgi
```

```
755
rwx
100 = 4
010 = 2
001 = 1
```

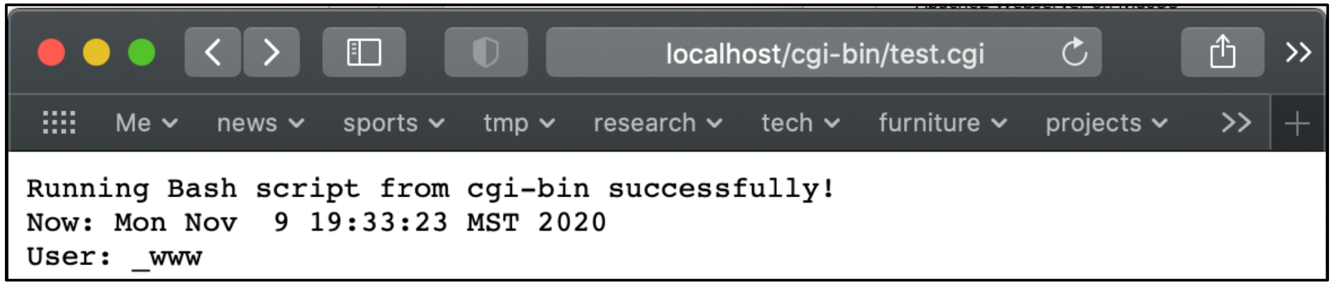
Apache2 Webserver on MacOS

- Simple bash script, let's call it **`test`**:

```
#!/bin/bash
echo "Content-Type: text/plain"
echo
echo "Running Bash script from cgi-bin successfully!"
echo -n "Now: "
date
echo -n "User: "
whoami
exit 0
```

Boilerplate browser expects:

```
Content-Type: text/plain
<blank line>
```

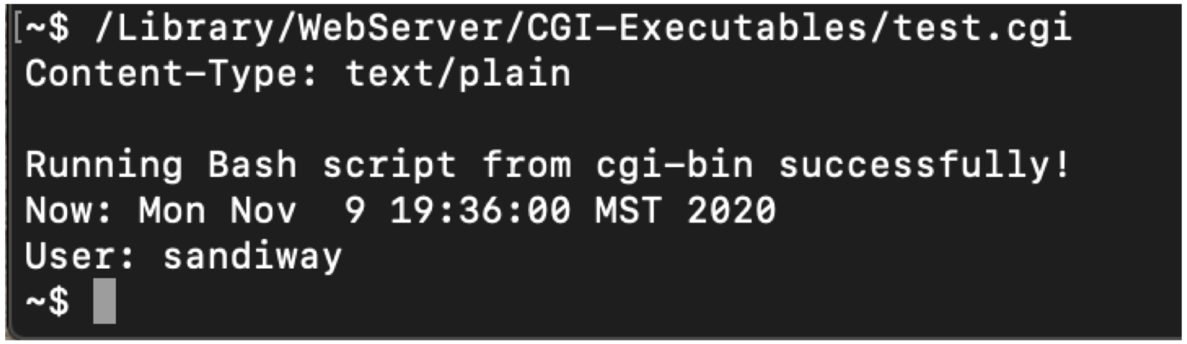


```
Running Bash script from cgi-bin successfully!
Now: Mon Nov  9 19:33:23 MST 2020
User: _www
```

making sure our bash script runs on the server…

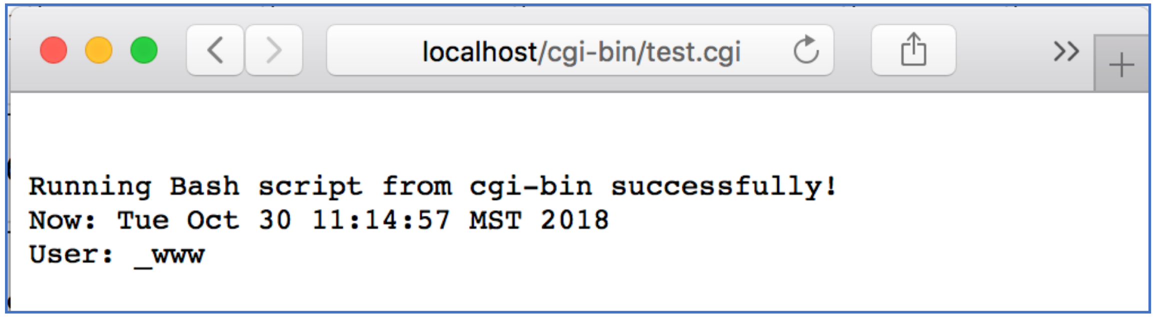
Apache2 Webserver on MacOS

Compare (running the bash shell script manually) on the command line:



```
[~$ /Library/WebServer/CGI-Executables/test.cgi
Content-Type: text/plain

Running Bash script from cgi-bin successfully!
Now: Mon Nov  9 19:36:00 MST 2020
User: sandiway
~$
```

`http://localhost/cgi-bin/test.cgi`

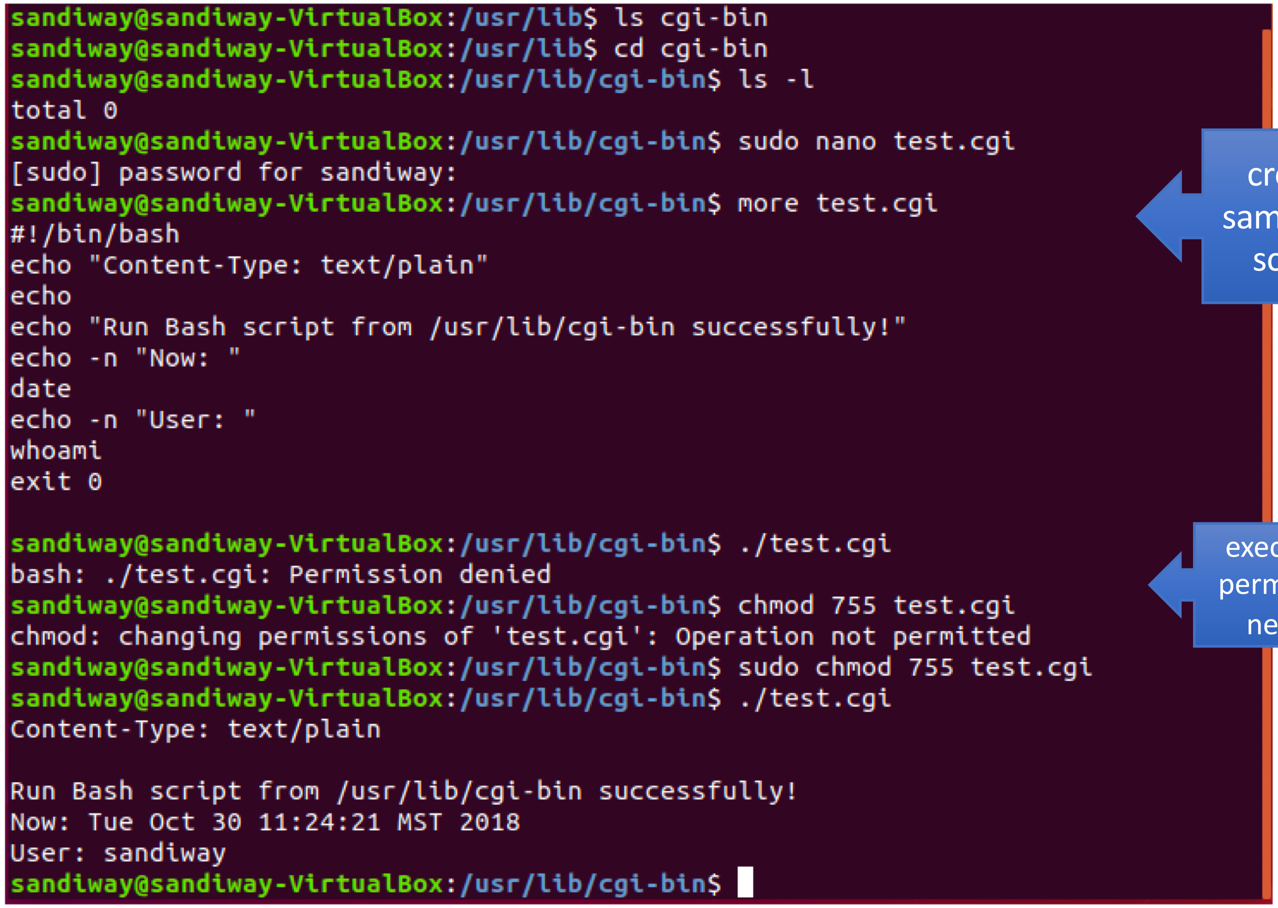


```
Running Bash script from cgi-bin successfully!
Now: Tue Oct 30 11:14:57 MST 2018
User: _www
```

- User:
- Preamble: content-type and blank line

Apache Webserver on Ubuntu

- http://localhost/cgi-bin/
- CGI binaries directory: **`/usr/lib/cgi-bin/`**
  - files must be made executable!



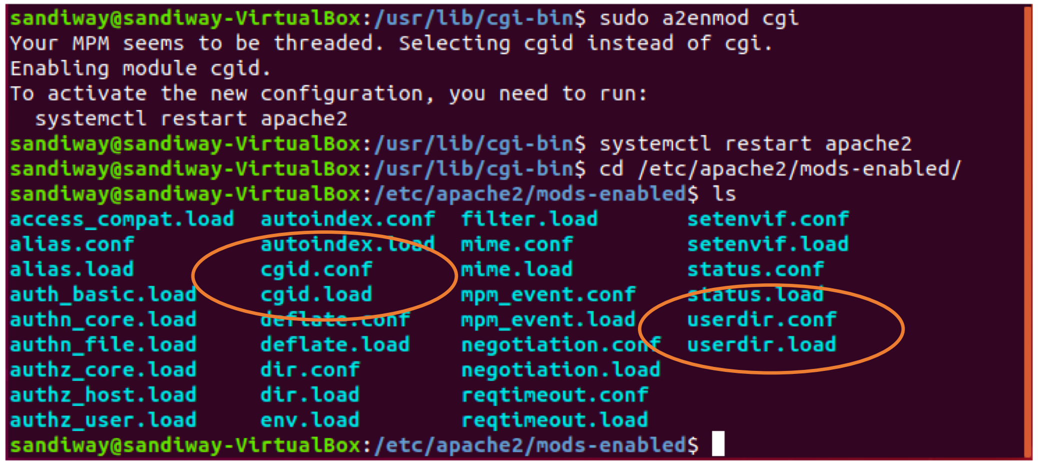
```
sandiway@sandiway-VirtualBox:/usr/lib$ ls cgi-bin
sandiway@sandiway-VirtualBox:/usr/lib$ cd cgi-bin
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ ls -l
total 0
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ sudo nano test.cgi
[sudo] password for sandiway:
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ more test.cgi
#!/bin/bash
echo "Content-Type: text/plain"
echo
echo "Run Bash script from /usr/lib/cgi-bin successfully!"
echo -n "Now: "
date
echo -n "User: "
whoami
exit 0

sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ ./test.cgi
bash: ./test.cgi: Permission denied
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ chmod 755 test.cgi
chmod: changing permissions of 'test.cgi': Operation not permitted
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ sudo chmod 755 test.cgi
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ ./test.cgi
Content-Type: text/plain

Run Bash script from /usr/lib/cgi-bin successfully!
Now: Tue Oct 30 11:24:21 MST 2018
User: sandiway
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$
```

create same test script

executable permissions needed

Apache Webserver on Ubuntu

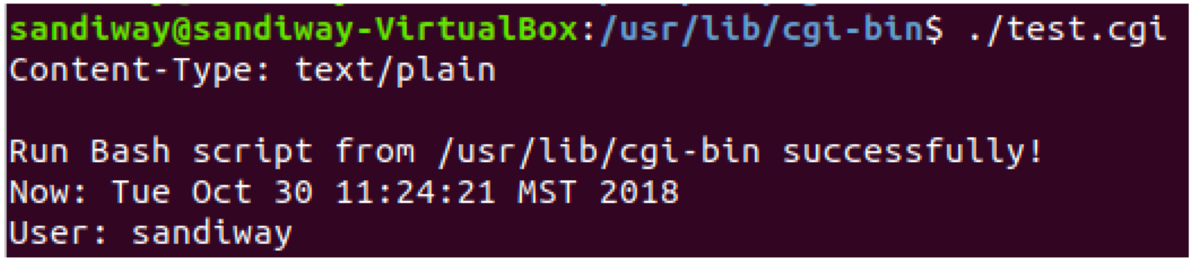
- Enabling cgi-bin:
  - **`sudo a2enmod cgi`** (*enables cgid instead of cgi*)
  - directory **`/etc/apache2/mods-enabled/`**



```
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ sudo a2enmod cgi
Your MPM seems to be threaded. Selecting cgid instead of cgi.
Enabling module cgid.
To activate the new configuration, you need to run:
  systemctl restart apache2
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ systemctl restart apache2
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ cd /etc/apache2/mods-enabled/
sandiway@sandiway-VirtualBox:/etc/apache2/mods-enabled$ ls
access_compat.load  autoindex.conf  filter.load       setenvif.conf
alias.conf          autoindex.load  mime.conf         setenvif.load
alias.load          cgid.conf       mime.load         status.conf
auth_basic.load     cgid.load       mpm_event.conf    status.load
authn_core.load     deflate.conf    mpm_event.load    userdir.conf
authn_file.load     deflate.load    negotiation.conf  userdir.load
authz_core.load     dir.conf        negotiation.load
authz_host.load     dir.load        reqtimeout.conf
authz_user.load     env.load        reqtimeout.load
sandiway@sandiway-VirtualBox:/etc/apache2/mods-enabled$
```

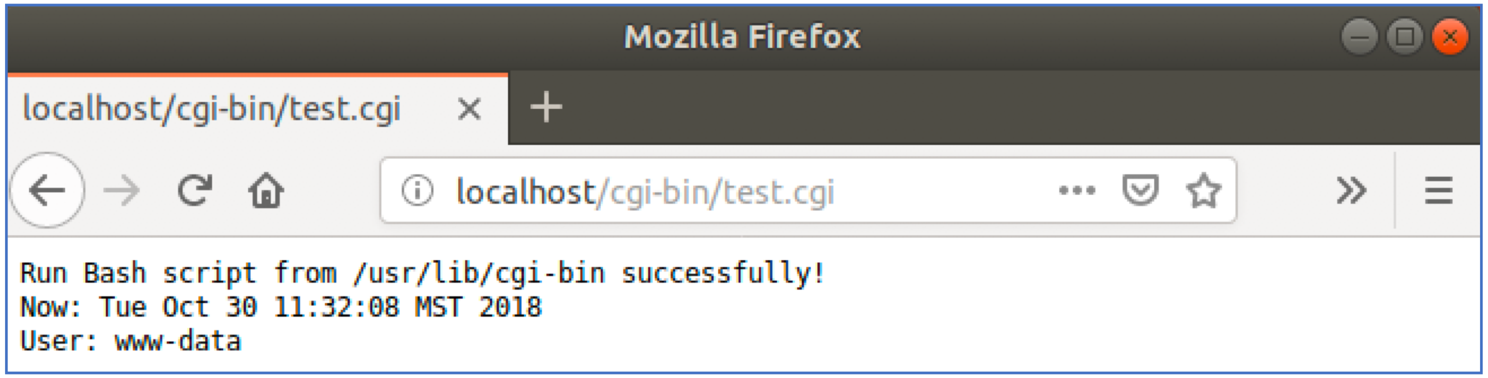
Apache Webserver on Ubuntu

- Compare running **`test`** directly:



```
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ ./test.cgi
Content-Type: text/plain

Run Bash script from /usr/lib/cgi-bin successfully!
Now: Tue Oct 30 11:24:21 MST 2018
User: sandiway
```

- **`http//localhost/cgi-bin/test:`**



```
Run Bash script from /usr/lib/cgi-bin successfully!
Now: Tue Oct 30 11:32:08 MST 2018
User: www-data
```

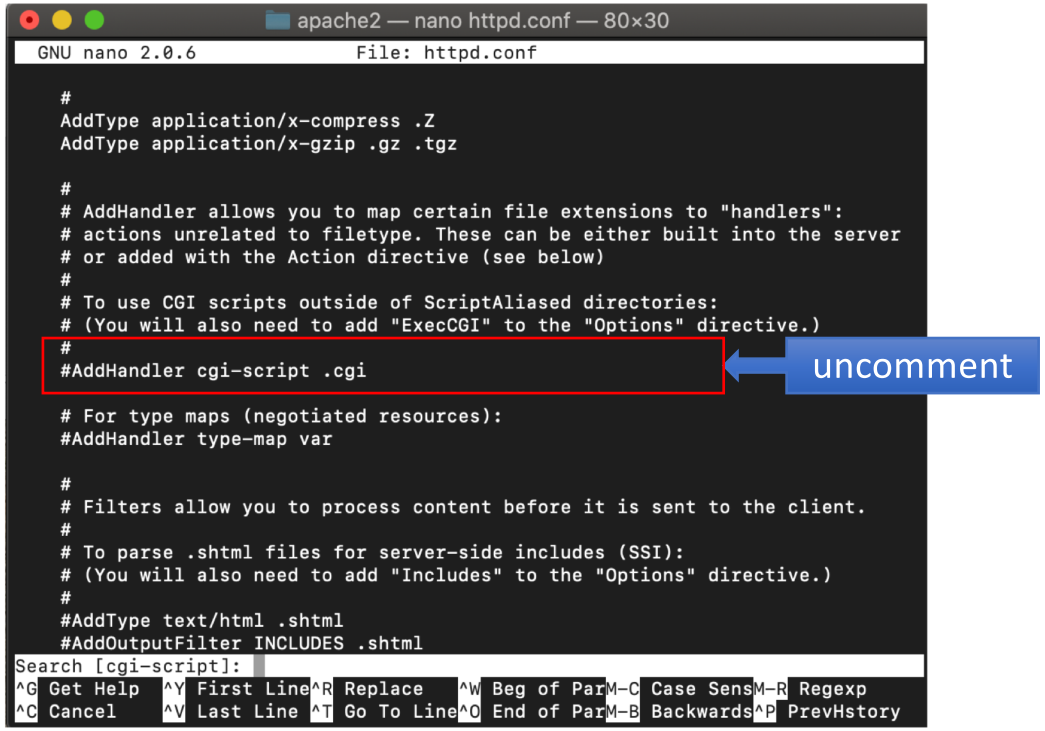
User: **`www-data`**
On OS X: **`_www`**

Running cgi-bin for users

Apache2 Webserver on MacOS

- To run programs in `~/Sites`, i.e. outside of `/Library/WebServer/CGI-Executables`
  - modify the Apache httpd configuration file:
  - **`sudo nano /etc/apache2/httpd.conf`**

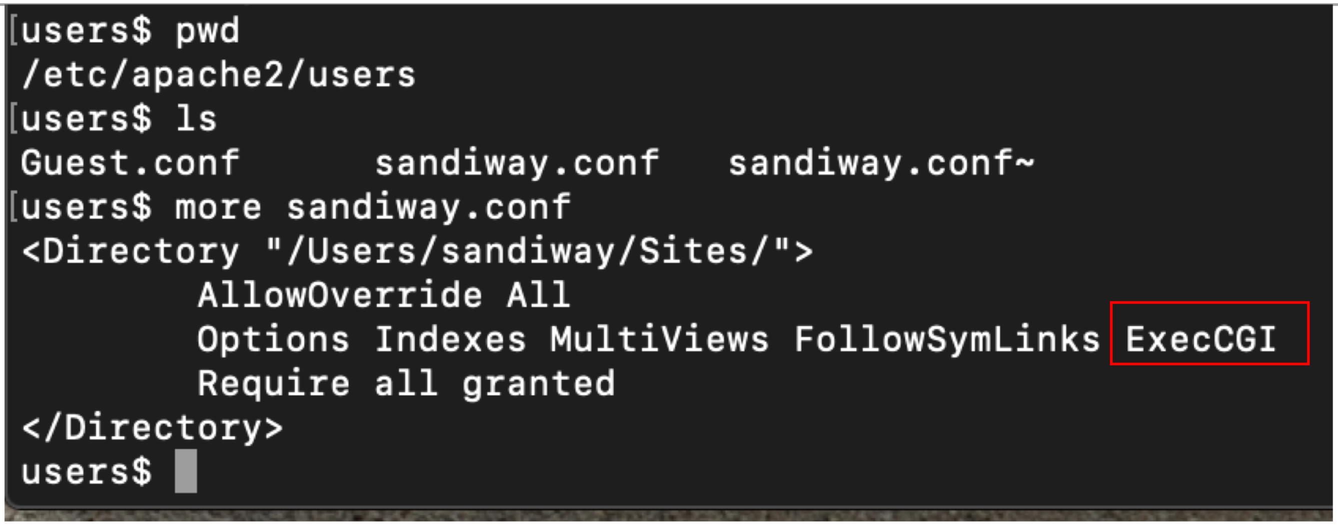
- *invokes the cgi-script handler for all files of type .cgi*



```
  GNU nano 2.0.6                File: httpd.conf

    #
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz

    #
    # AddHandler allows you to map certain file extensions to "handlers":
    # actions unrelated to filetype. These can be either built into the server
    # or added with the Action directive (see below)
    #
    # To use CGI scripts outside of ScriptAliased directories:
    # (You will also need to add "ExecCGI" to the "Options" directive.)
    #
    #AddHandler cgi-script .cgi

    # For type maps (negotiated resources):
    #AddHandler type-map var

    #
    # Filters allow you to process content before it is sent to the client.
    #
    # To parse .shtml files for server-side includes (SSI):
    # (You will also need to add "Includes" to the "Options" directive.)
    #
    #AddType text/html .shtml
    #AddOutputFilter INCLUDES .shtml
Search [cgi-script]:
^G Get Help   ^Y First Line^R Replace    ^W Beg of ParM-C Case SensM-R Regexp
^C Cancel     ^V Last Line ^T Go To Line^O End of ParM-B Backwards^P PrevHstory
```

Apache2 Webserver on MacOS

- Also modify **`/etc/apache2/users/sandiway.conf`**
  - *already created in last lecture…*
  - to add the ExecCGI option as follows:



```
[users$ pwd
/etc/apache2/users
[users$ ls
Guest.conf      sandiway.conf   sandiway.conf~
[users$ more sandiway.conf
<Directory "/Users/sandiway/Sites/">
        AllowOverride All
        Options Indexes MultiViews FollowSymLinks ExecCGI
        Require all granted
</Directory>
users$ 
```

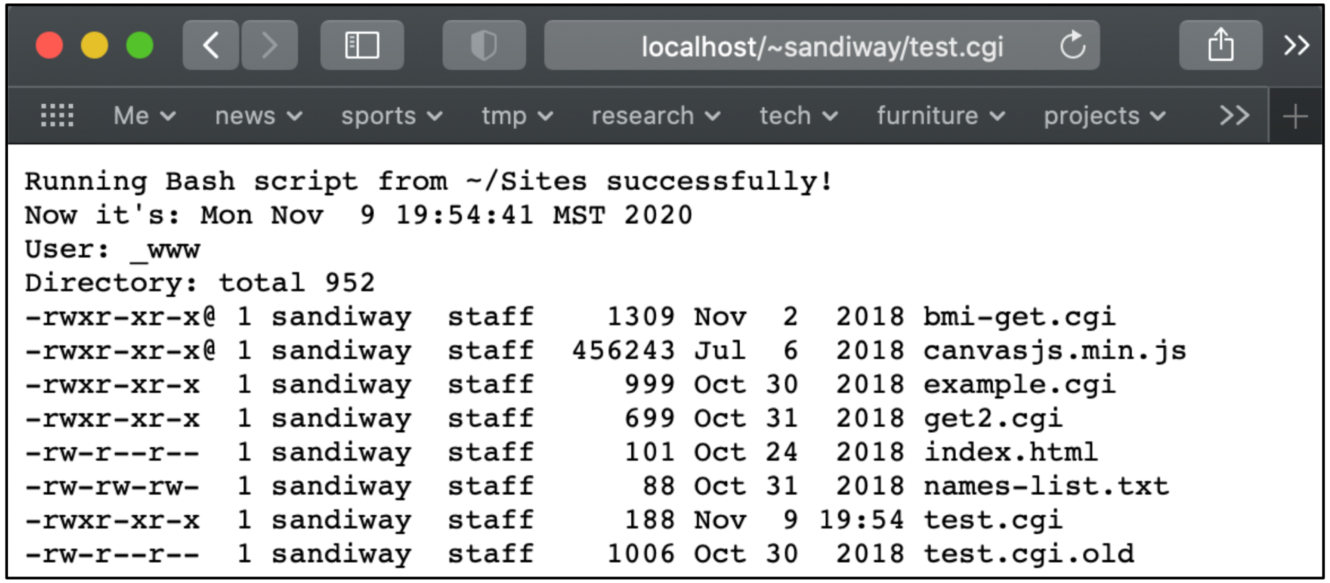
Apache2 Webserver on MacOS

- File **`~/Sites/test.cgi`**

```
#!/bin/bash
echo "Content-Type: text/plain"
echo
echo "Running Bash script from ~sandiway/Sites successfully!"
echo -n "Now: "
date
echo -n "User: "
whoami
echo -n "Directory: "
ls -l
exit 0
```

Apache2 Webserver on MacOS

- sudo apachectl -k restart
- **http://localhost/~sandiway/test.cgi**

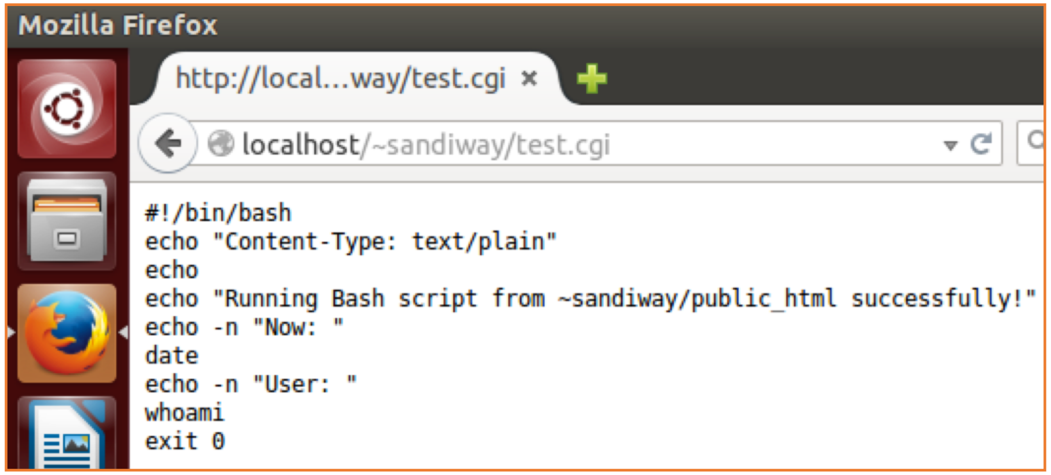


```
Running Bash script from ~/Sites successfully!
Now it's: Mon Nov  9 19:54:41 MST 2020
User: _www
Directory: total 952
-rwxr-xr-x@ 1 sandiway  staff    1309 Nov  2  2018 bmi-get.cgi
-rwxr-xr-x@ 1 sandiway  staff  456243 Jul  6  2018 canvasjs.min.js
-rwxr-xr-x  1 sandiway  staff     999 Oct 30  2018 example.cgi
-rwxr-xr-x  1 sandiway  staff     699 Oct 31  2018 get2.cgi
-rw-r--r--  1 sandiway  staff     101 Oct 24  2018 index.html
-rw-rw-rw-  1 sandiway  staff      88 Oct 31  2018 names-list.txt
-rwxr-xr-x  1 sandiway  staff     188 Nov  9 19:54 test.cgi
-rw-r--r--  1 sandiway  staff    1006 Oct 30  2018 test.cgi.old
```

Apache2 Webserver on Ubuntu

- https://httpd.apache.org/docs/2.4/howto/cgi.html
- By default cgi-bin is not enabled for **`~/public_html`**

just displays program instead of running it



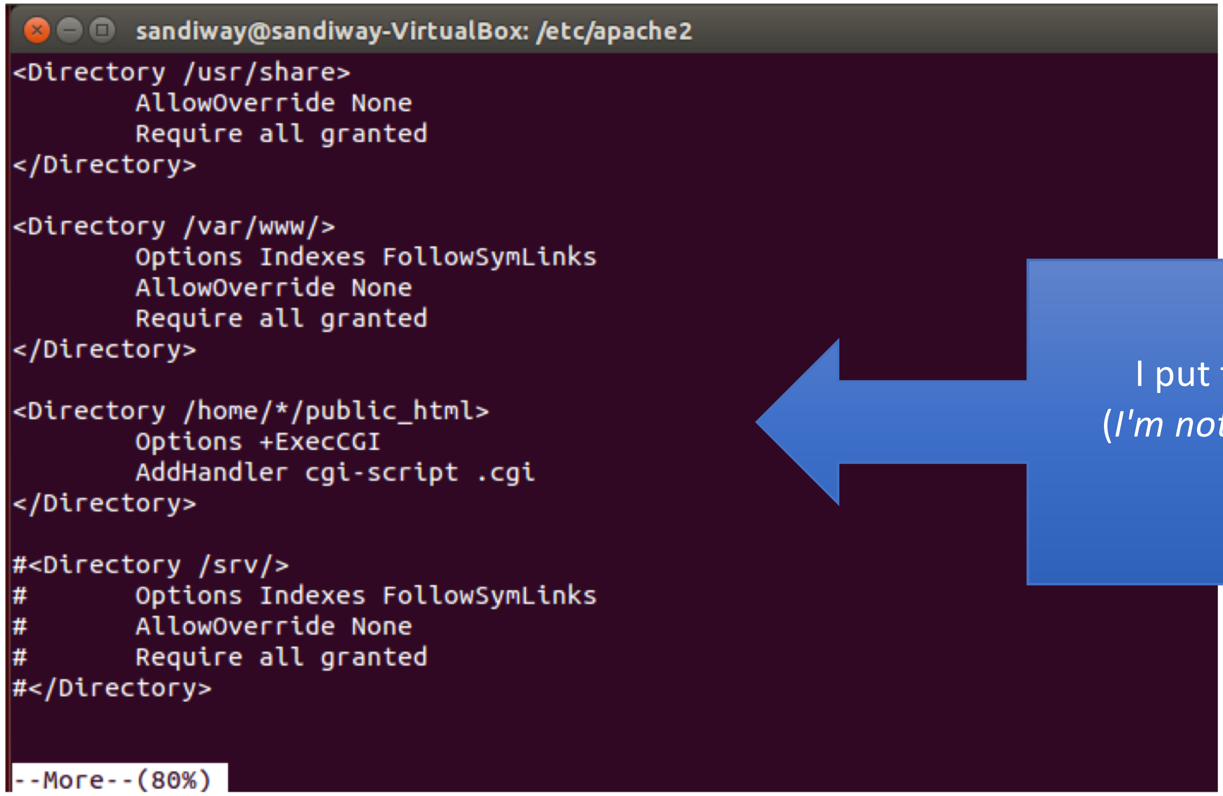
```
#!/bin/bash
echo "Content-Type: text/plain"
echo
echo "Running Bash script from ~sandiway/public_html successfully!"
echo -n "Now: "
date
echo -n "User: "
whoami
exit 0
```

Apache Webserver on Ubuntu

- From https://httpd.apache.org/docs/current/howto/cgi.html
  - add these lines to /etc/apache2/apache.conf
    - `<Directory "/home/*/public_html">`
    - `    Options +ExecCGI`
    - `    AddHandler cgi-script .cgi`
    - `</Directory>`
  - and restart apache2:
    - `sudo systemctl restart apache2`

Apache Webserver on Ubuntu

- **`/etc/apache2/apache2.conf`**

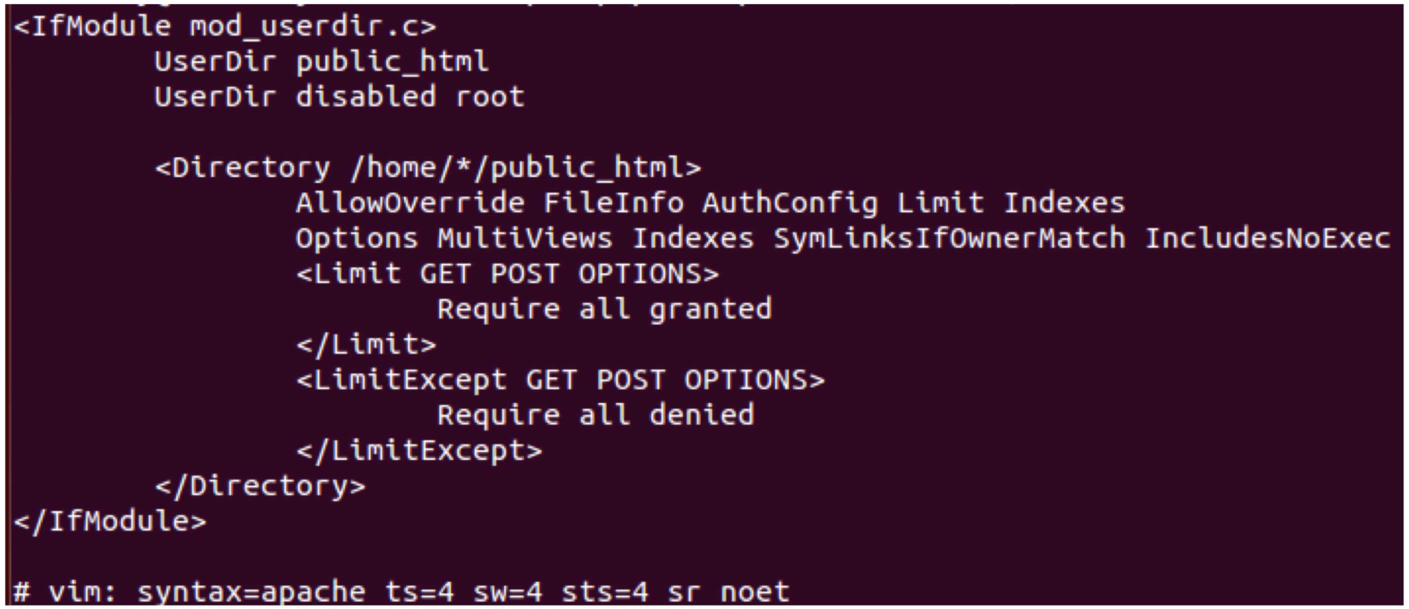


```
<Directory /usr/share>
        AllowOverride None
        Require all granted
</Directory>

<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>

<Directory /home/*/public_html>
        Options +ExecCGI
        AddHandler cgi-script .cgi
</Directory>

#<Directory /srv/>
#       Options Indexes FollowSymLinks
#       AllowOverride None
#       Require all granted
#</Directory>

--More--(80%)
```

I put the declarations here
(I'm not sure it's the best place though…)

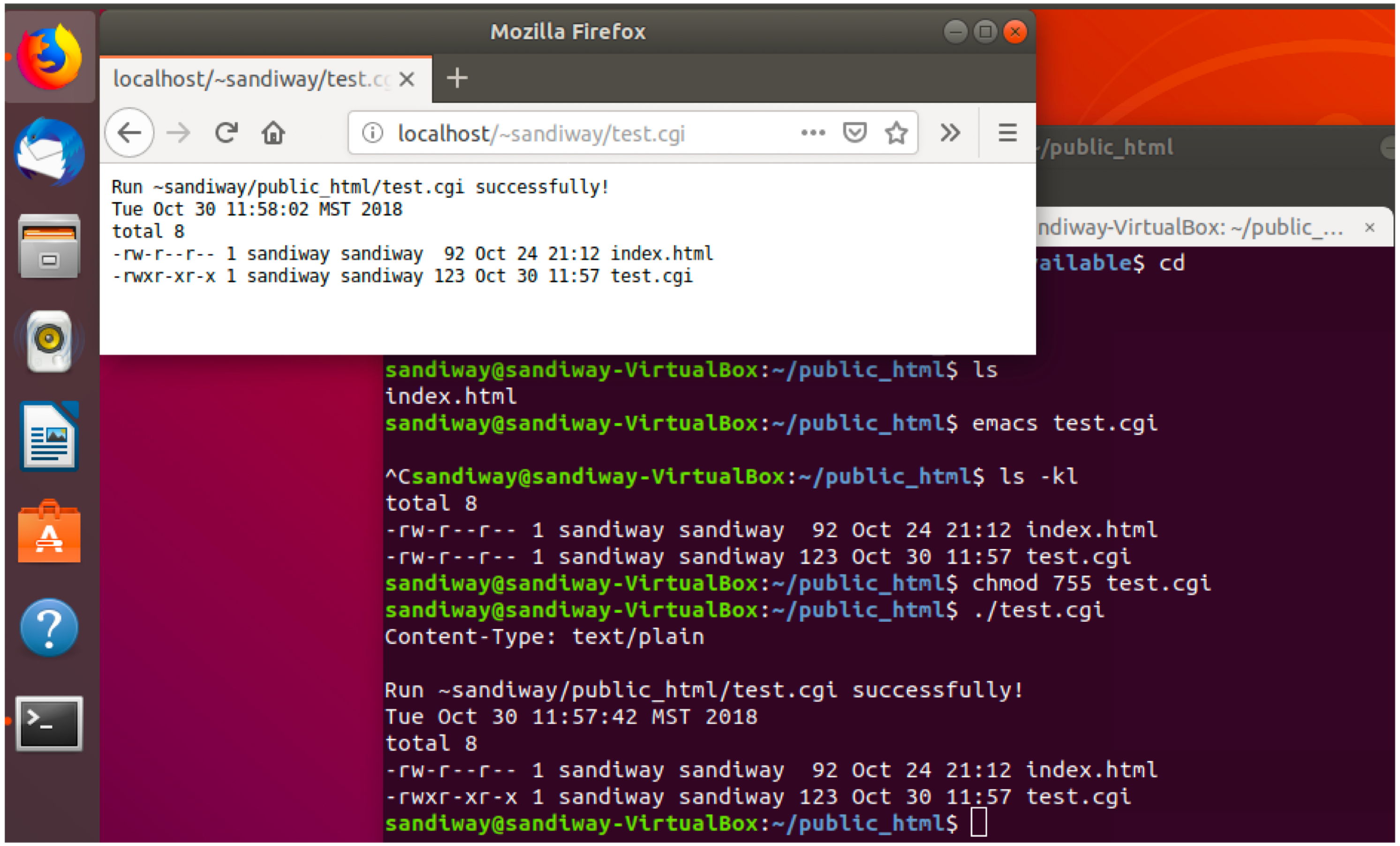
Apache Webserver on Ubuntu

- **`/etc/apache2/mods-available/userdir.conf`**



```
<IfModule mod_userdir.c>
        UserDir public_html
        UserDir disabled root

        <Directory /home/*/public_html>
                AllowOverride FileInfo AuthConfig Limit Indexes
                Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
                <Limit GET POST OPTIONS>
                        Require all granted
                </Limit>
                <LimitExcept GET POST OPTIONS>
                        Require all denied
                </LimitExcept>
        </Directory>
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Apache Webserver on Ubuntu

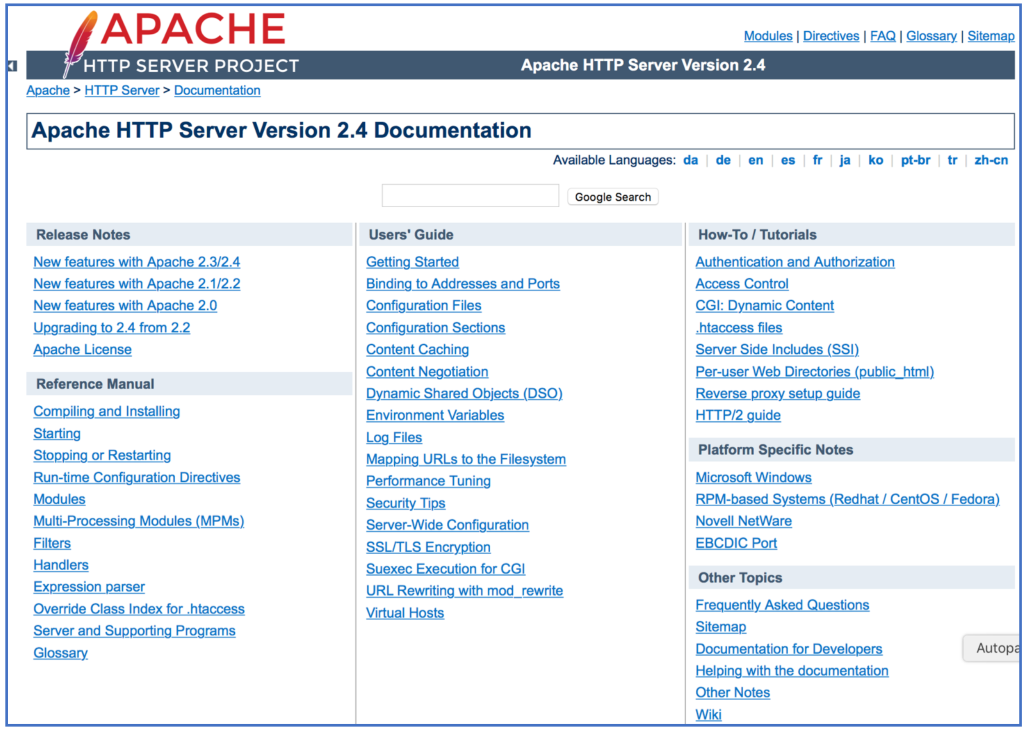

Mozilla Firefox
localhost/~sandiway/test.cg
localhost/~sandiway/test.cgi
~/public_html
ndiway-VirtualBox: ~/public_...

```
Run ~sandiway/public_html/test.cgi successfully!
Tue Oct 30 11:58:02 MST 2018
total 8
-rw-r--r-- 1 sandiway sandiway  92 Oct 24 21:12 index.html
-rwxr-xr-x 1 sandiway sandiway 123 Oct 30 11:57 test.cgi
```

```
ailable$ cd
sandiway@sandiway-VirtualBox:~/public_html$ ls
index.html
sandiway@sandiway-VirtualBox:~/public_html$ emacs test.cgi

^Csandiway@sandiway-VirtualBox:~/public_html$ ls -kl
total 8
-rw-r--r-- 1 sandiway sandiway  92 Oct 24 21:12 index.html
-rw-r--r-- 1 sandiway sandiway 123 Oct 30 11:57 test.cgi
sandiway@sandiway-VirtualBox:~/public_html$ chmod 755 test.cgi
sandiway@sandiway-VirtualBox:~/public_html$ ./test.cgi
Content-Type: text/plain

Run ~sandiway/public_html/test.cgi successfully!
Tue Oct 30 11:57:42 MST 2018
total 8
-rw-r--r-- 1 sandiway sandiway  92 Oct 24 21:12 index.html
-rwxr-xr-x 1 sandiway sandiway 123 Oct 30 11:57 test.cgi
sandiway@sandiway-VirtualBox:~/public_html$
```

Documentation

`http://httpd.apache.org/docs/current/`



Modules | Directives | FAQ | Glossary | Sitemap

Apache HTTP Server Version 2.4

Apache > HTTP Server > Documentation

Apache HTTP Server Version 2.4 Documentation

Available Languages: da | de | en | es | fr | ja | ko | pt-br | tr | zh-cn

Release Notes

- New features with Apache 2.3/2.4
- New features with Apache 2.1/2.2
- New features with Apache 2.0
- Upgrading to 2.4 from 2.2
- Apache License

Reference Manual

- Compiling and Installing
- Starting
- Stopping or Restarting
- Run-time Configuration Directives
- Modules
- Multi-Processing Modules (MPMs)
- Filters
- Handlers
- Expression parser
- Override Class Index for .htaccess
- Server and Supporting Programs
- Glossary

Users' Guide

- Getting Started
- Binding to Addresses and Ports
- Configuration Files
- Configuration Sections
- Content Caching
- Content Negotiation
- Dynamic Shared Objects (DSO)
- Environment Variables
- Log Files
- Mapping URLs to the Filesystem
- Performance Tuning
- Security Tips
- Server-Wide Configuration
- SSL/TLS Encryption
- Suexec Execution for CGI
- URL Rewriting with mod_rewrite
- Virtual Hosts

How-To / Tutorials

- Authentication and Authorization
- Access Control
- CGI: Dynamic Content
- .htaccess files
- Server Side Includes (SSI)
- Per-user Web Directories (public_html)
- Reverse proxy setup guide
- HTTP/2 guide

Platform Specific Notes

- Microsoft Windows
- RPM-based Systems (Redhat / CentOS / Fedora)
- Novell NetWare
- EBCDIC Port

Other Topics

- Frequently Asked Questions
- Sitemap
- Documentation for Developers
- Helping with the documentation
- Other Notes
- Wiki

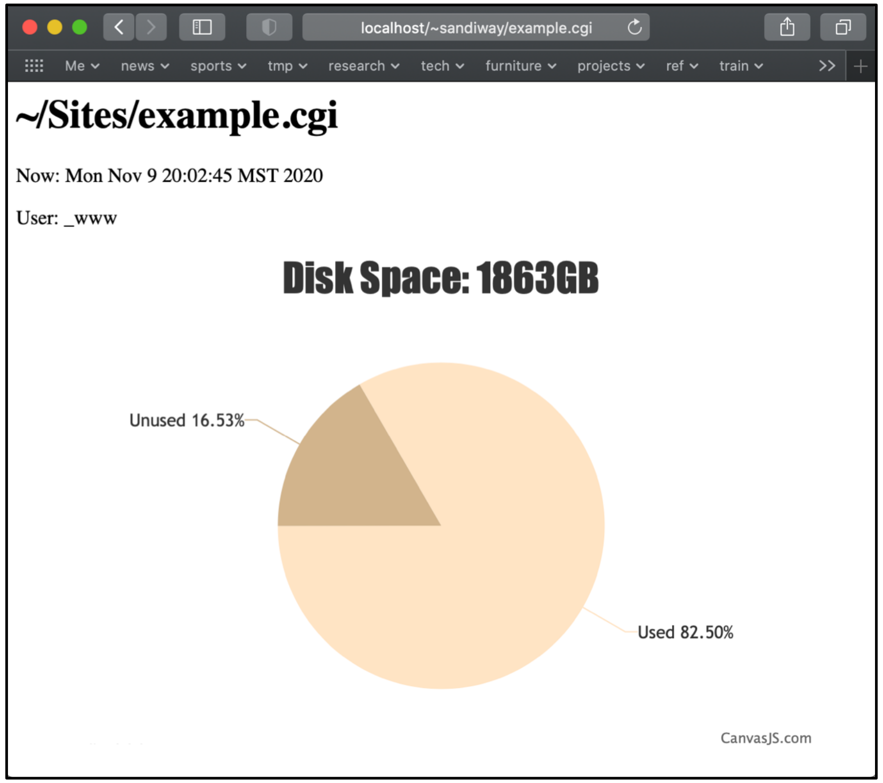
Homework 9

- Make up two **different** bash shell scripts, one for the root server and one for the user.

[Refresh your knowledge of bash scripting from the first few lectures.]

- Show them running using `localhost/cgi-bin/example.cgi` and `localhost/~user/example.cgi`
- Send me screen snapshots.
- Be adventurous!
- Due date next Monday midnight

An Example using HTML/Javascript

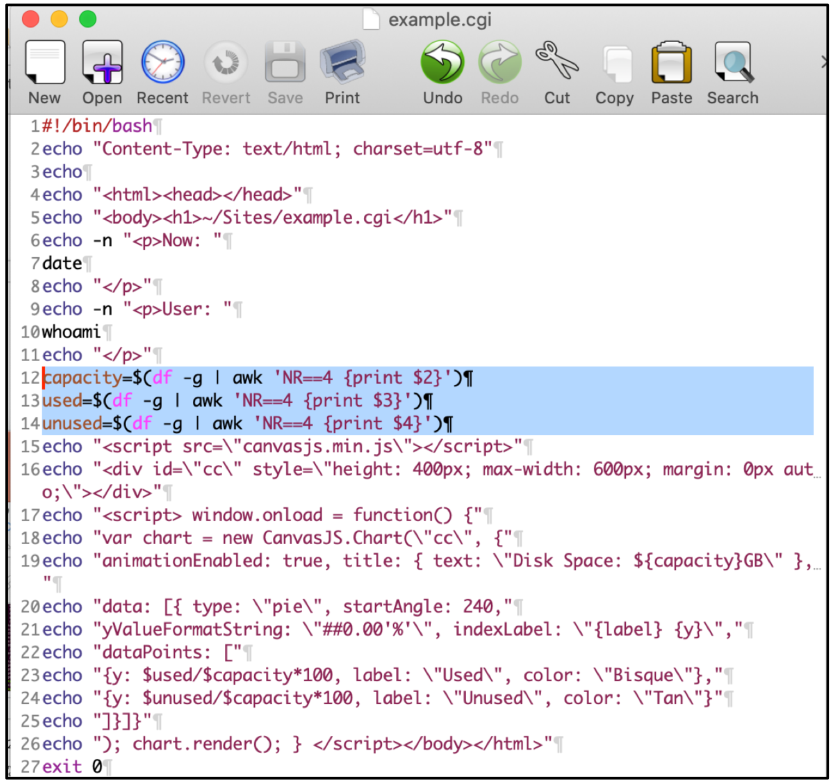

localhost/~sandiway/example.cgi
Me news sports tmp research tech furniture projects ref train

~/Sites/example.cgi

Now: Mon Nov 9 20:02:45 MST 2020

User: _www

Disk Space: 1863GB
Unused 16.53%
Used 82.50%
CanvasJS.com


example.cgi
New Open Recent Revert Save Print Undo Redo Cut Copy Paste Search

```
#!/bin/bash
echo "Content-Type: text/html; charset=utf-8"
echo
echo "<html><head></head>"
echo "<body><h1>~/Sites/example.cgi</h1>"
echo -n "<p>Now: "
date
echo "</p>"
echo -n "<p>User: "
whoami
echo "</p>"
capacity=$(df -g | awk 'NR==4 {print $2}')
used=$(df -g | awk 'NR==4 {print $3}')
unused=$(df -g | awk 'NR==4 {print $4}')
echo "<script src=\"canvasjs.min.js\"></script>"
echo "<div id=\"cc\" style=\"height: 400px; max-width: 600px; margin: 0px auto;\"></div>"
echo "<script> window.onload = function() {"
echo "var chart = new CanvasJS.Chart(\"cc\", {"
echo "animationEnabled: true, title: { text: \"Disk Space: ${capacity}GB\" },"
echo "data: [{ type: \"pie\", startAngle: 240,"
echo "yValueFormatString: \"##0.00'%'\", indexLabel: \"{label} {y}\","
echo "dataPoints: ["
echo "{y: $used/$capacity*100, label: \"Used\", color: \"Bisque\"},"
echo "{y: $unused/$capacity*100, label: \"Unused\", color: \"Tan\"}"
echo "]}]}"
echo "); chart.render(); } </script></body></html>"
exit 0
```