

1 Explain how to create CGI bin directory:

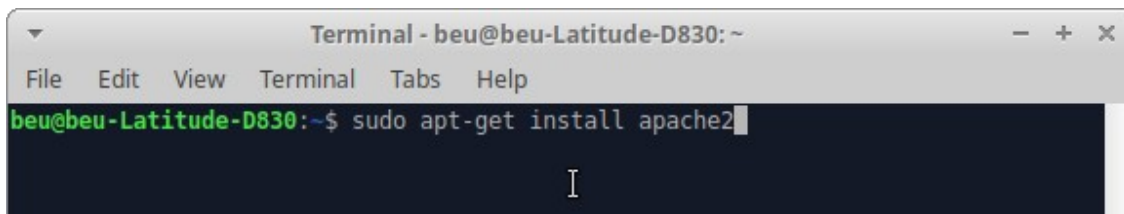
CGI stands for common gateway interface basically for a standard that provides an external gateway to interface servers.

Prerequisites: System we are working on should have an access to some local server. Here I am using apache2.

The following command will install the latest meta package apache2.

sudo apt update

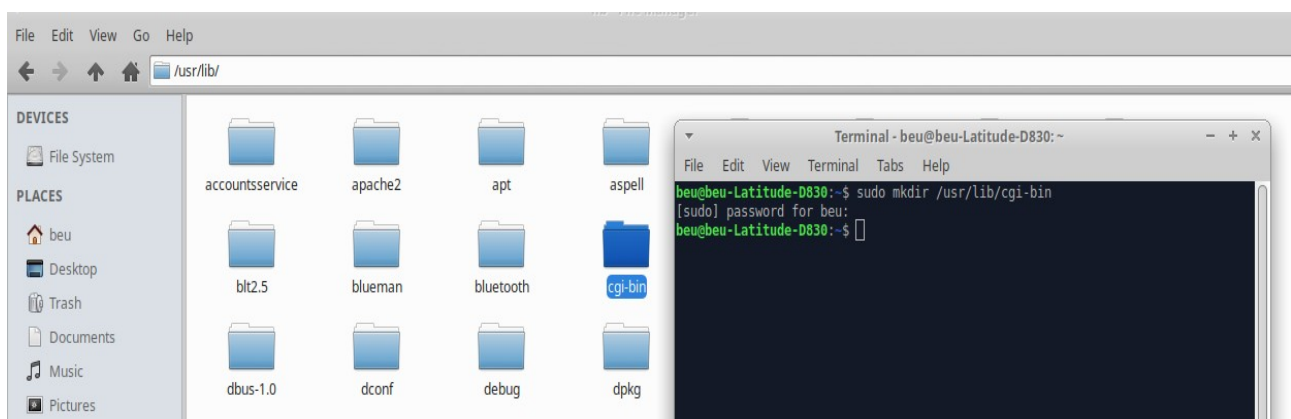
sudo apt install apache2

A terminal window titled "Terminal - beu@beu-Latitude-D830: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command prompt shows "beu@beu-Latitude-D830:~\$ sudo apt-get install apache2" with a cursor at the end.

Goal: Creating a CGI bin directory and allow CGI to get the apache server to recognize the directory which we have created and to allow the same for the execution of commands from within that directory.

Construction:

Step 1: Creating it with the command ***sudo mkdir /usr/lib/cgi-bin*** would create a **cgi-bin** directory within the **/usr/lib/** directory. This is where our Python programs and other various files will be placed. Generally, it is the repository for our executables.



Step 2: Issuing the ***sudo chmod 755 /usr/lib/cgi-bin*** command to provide write permission only for the owner and read/execute permission for others and ***sudo chown root.root /usr/lib/cgi-bin*** to sets the user and group of cgi-bin to root.

Step 3: The command `ls -l /usr/lib` can be used to check our directory's permissions.

```
Terminal - beu@beu-Latitude-D830: ~
File Edit View Terminal Tabs Help
beu@beu-Latitude-D830:~$ sudo chmod 755 /usr/lib/cgi-bin
[sudo] password for beu:
beu@beu-Latitude-D830:~$ sudo chown root.root /usr/lib/cgi-bin
beu@beu-Latitude-D830:~$ ls -l /usr/lib
total 2728
drwxr-xr-x 2 root root 4096 Feb 3 23:55 accountsservice
drwxr-xr-x 3 root root 4096 May 1 23:29 apache2
drwxr-xr-x 5 root root 4096 Feb 3 23:53 apt
drwxr-xr-x 3 root root 4096 Feb 3 23:56 aspell
drwxr-xr-x 2 root root 4096 Feb 3 23:56 atril
drwxr-xr-x 2 root root 4096 Feb 3 23:56 at-spi2-core
drwxr-xr-x 2 root root 4096 Feb 3 23:56 avahi
drwxr-xr-x 2 root root 4096 Apr 20 2018 binfmt.d
drwxr-xr-x 2 root root 4096 May 1 12:51 blt2.5
drwxr-xr-x 2 root root 4096 Feb 3 23:56 bluedevil
drwxr-xr-x 2 root root 4096 Apr 29 22:24 bluetooth
drwxr-xr-x 2 root root 4096 May 3 13:32 cgi-bin
drwxr-xr-x 2 root root 4096 Apr 29 22:11 chromium-browser
-rwxr-xr-x 1 root root 684 May 5 2018 cnf-update-db
drwxr-xr-x 2 root root 4096 Feb 3 23:56 colord
-rwxr-xr-x 1 root root 3300 May 5 2018 command-not-found
drwxr-xr-x 2 root root 4096 Apr 29 22:24 compat-ld
drwxr-xr-x 10 root root 4096 Feb 3 23:56 cups
drwxr-xr-x 2 root root 4096 Feb 3 23:53 dbus-1.0
```

Step 4: Activating the VirtualHost Configuration File by adding the following line in the file `/etc/apache2/sites-available/000-default.conf`

```
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
AllowOverride None
Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
Order allow,deny
Allow from all
</Directory>
```

By doing so, apache will now know about CGI and where its directories are located.

```
/etc/apache2/sites-available/000-default.conf - Mousepad
File Edit Search View Document Help
1 <VirtualHost *:80>
2 # The ServerName directive sets the request scheme, hostname and port that
3 # the server uses to identify itself. This is used when creating
4 # redirection URLs. In the context of virtual hosts, the ServerName
5 # specifies what hostname must appear in the request's Host: header to
6 # match this virtual host. For the default virtual host (this file) this
7 # value is not decisive as it is used as a last resort host regardless.
8 # However, you must set it for any further virtual host explicitly.
9 #ServerName www.example.com
10
11 ServerAdmin webmaster@localhost
12 DocumentRoot /var/www/html
13
14 # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
15 # error, crit, alert, emerg.
16 # It is also possible to configure the loglevel for particular
17 # modules, e.g.
18 #LogLevel info ssl:warn
19
20 ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
21 <Directory "/usr/lib/cgi-bin">
22 AllowOverride None
23 Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
24 Order allow,deny
25 Allow from all
26 </Directory>
27
28 ErrorLog ${APACHE_LOG_DIR}/error.log
29 CustomLog ${APACHE_LOG_DIR}/access.log combined
30
31 # For most configuration files from conf-available/, which are
32 # enabled or disabled at a global level, it is possible to
33 # include a line for only one particular virtual host. For example the
34 # following line enables the CGI configuration for this host only
35 # after it has been globally disabled with "a2disconf".
36 #Include conf-available/serve-cgi-bin.conf
37 </VirtualHost>
38
39 # vim: syntax=apache ts=4 sw=4 sts=4 sr noet
40
```

Now our cgi-bin directory is ready.

2 Implement the hello world program in CGI fashion using python.

Step 1 : Creating a python script as following.

```
#!/usr/bin/python

print("content-type:text/html\r\n\r\n")

print("<html>")

print("<head>")

print("<title>My very first CGI</title>")

print("</head>")

print("<body>")

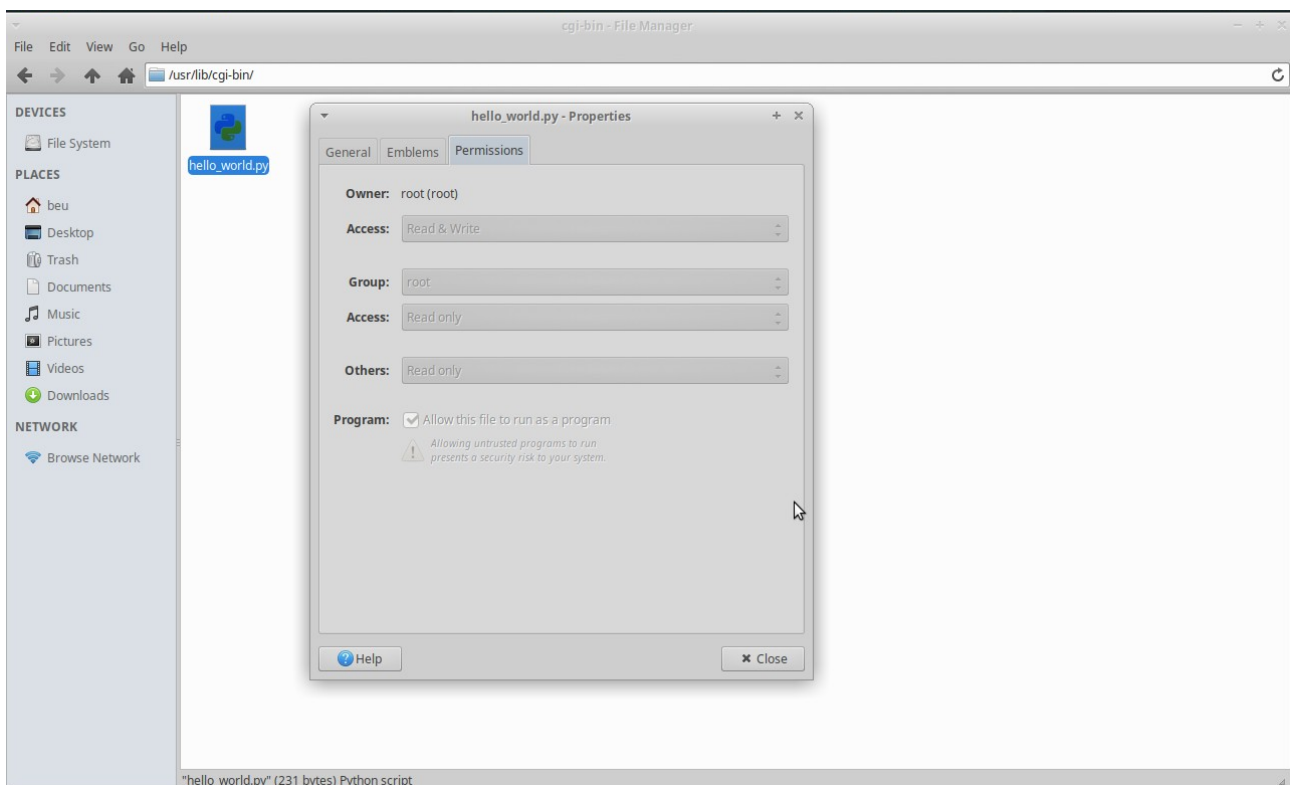
print("<h1>HELLO WORLD !</h1>")

print("</body>")

print("</html>")
```

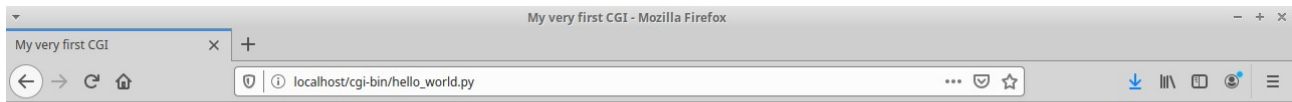
Step 2 : Saving the above python script file in /usr/lib/cgi-bin as hello_world.py and giving it 755 permissions (means read and execute access for everyone and also write access for the owner of the file).

This can be achieved by giving the command **sudo chmod -R 755 /usr/lib/cgi-bin/hello_world.py**



Step 3: Start apache2 server by issuing the command **sudo service apache2 start**.

Step 4: Opening that file in web browser by pointing it to **http://localhost/cgi-bin/hello_world.py**



HELLO WORLD !