

**1. What is HTTP?**

A network protocol, which is for fetching resources such as HTML documents.

**2. What is HTML?**

A markup language used for constructing web page and its content.

**3. What is CGI?**

CGI is the abbreviation of computer-generated imagery, which is the application of computer graphics to create or contribute to images in art, printed media, etc.

**4. List 15 Unix commands. (Show how they are used.)**

du: estimate file usage

cd: go to directory

ls: list files in under current path

cp: copy files and directories

chmod: change the access permissions of file system objects

clear: clear the terminal screen

mkdir: make a dictionary under current path

rmdir: remove an empty directory

rm: remove objects such as files

gunzip: compress or expand a file or a list of files

mv: move and rename files and directories

pwd: print the path of the working directory, starting from the root

grep: search plain-text data sets for lines that match a regular expression

netstat: display network connections for TCP, routing tables, and a number of network interface and network protocols statistics

cat: read files sequentially and write them to standard output.

**5. What is Apache?**

A website server and a foundation. Apache server is a web server application which can deliver content such as HTML pages, multimedia and CSS style sheets over the internet. It is responsible for accepting directory HTTP requests from internet users and sending them the information they want in the form of files and web pages.

**6. List and briefly describe the Apache directory structure. (Say a few words about each one of the folders mentioned.)**

/u/apache/htdocs/index.html	(default homepage)
/one.php	(php information)
/protected/hw1.html	(protected folder)
/phpsessions	(folder to store phpsessions)
/passwd	(password)
/cgi-bin/one	(CGI-BIN folder)
/logs/access_log	(access record)
/error_log	(error record)

/httpd.pid	(running pid)
/conf/httpd.conf	(file configuration)
/php.ini	(php configuration)
/bin/apachectl	(start the apache server)
/htpasswd	(bin program for setup password)

**7. Describe four ways in which you can determine if your Apache server is up or not.**

We can check the status of website in the browser, or we can use grep port, or we can use ps - ef | grep username, or we can use the telnet.

**8. How can you find out the process id of your Apache server (if it's running)?**

If it is running, it's in httpd.pid. But remember our NFS setup here.

**9. What is chmod and how do we use it?**

chmod is a Linux command which can make a file to be executable.

chmod +x filename

**10. How do you set up Apache to be restarted 4 times a day?**

Go to the Apache folder and enter command crontab -e, and then I can set up the entry as:

0 0,6,12,18 \* \* \* /u/duozwang/a348-workspace/apache/bin/apachectl restart

This means my server will automatically restart four times a day, at 0:00, 6:00, 12:00, 18:00.

**11. Where have we used tail -f in this class?**

To look at changes in real time at the end of the ~/apache/logs/error\_log file. For example, when learning about restarting with crontab or after installing PHP to make sure it worked, although in that case you don't need -f, it's enough to put a number after the dash.

**12. What is netstat? What does netstat -a produce as output? (Give some examples, be more specific.)**

netstat is a Linux command which can display network connections for TCP, routing tables, and a number of network interface. It will show what's happening with the network interface, port by port. netstat -a will output all the usable ports.

**13. What are the steps and commands in the installation of Apache?**

Copy the compressed archive.

Uncompress, unarchive, move into folder.

make clean, ./configure, make, make install

Put personal port number on the line with Listen in httpd.conf

Start personal server (indicate the command, also how to stop it).

Set up personal home page.

**14. Same question for PHP?**

Copy the compressed archive.

Uncompress, unarchive, move into folder.  
make clean, ./configure, make, make install  
Move and update php.ini, create phpsessions folder  
Add lines for .php and .phps extensions to httpd.conf

**15. Same question for the MySQL server.**

Copy the compressed archive.  
Uncompress, unarchive, move into folder.  
make clean, ./configure, make, make install  
Create and run setup that sets up the data directories  
Create start, stop, connect\_as\_root  
Create database, user, give user access to database, connect as user.

**16. Differences in installation between: Apache and PHP.**

Apache is a standalone server and PHP is an add-on module.

**17. Steps unique to MySQL installation.**

Creating users, databases, giving access rights.  
Initially populating the database(s) is also unique, as is setup.

**18. Steps unique to Tomcat installation.**

Get the archive.  
Uncompress, unarchive it.  
Define server.xml  
Define .tomcat-settings  
Source it, start the server  
Create index.html

Relevant commands:  
cp ~/apache-tomcat-9.0.53.tar.gz .  
tar xvf apache-tomecat-9.0.53.tar

**19. How many times did you change httpd.conf and for what purpose?**

Ports for apache, PHP installation, protected.

**20. What role does php.ini serve. How do we get it. Where do we keep it.**

Generate by installation (configure, make, make install).  
We can keep it anywhere, we choose apache/conf for that.  
It tells PHP how to behave (path to session folder, ways to process input)

**21. Client-side vs. server side state: differences and similarities.**

Client-side: simple, vulnerable  
Server-side: sticky, less-vulnerable. Requires user ID.  
Both Client-side state and server side state can store data. The differences are stated above.

**22. How can you keep track of the number of reloads in a client side state program?**

No, we cannot.

**23. How do you set up a password protected folder in Apache?**

First, use `apache/bin/htpasswd` and create a folder with usernames and passwords (mine is in `apache/htdocs/`). Next, create a folder that you want protected (`apache/htdocs/protected`). Finally in `apache/conf/httpd.conf` add code indicating that this folder is accessible to only some users and that the `passwd` file has the passwords for those users.

**24. How else could you do that in Apache?**

One can also use `.htaccess` and/or via a script that first asks for username and password. `.htaccess` files allow users to configure directories of the web server they control without modifying the main configuration file.

**1. Write the shortest/simplest CGI/Perl program that keeps state:**

```
#!/usr/bin/perl
use CGI;
&q = new CGI;
$times = $q->param("times")
$message = $q->param("message")
$action = $q->param("action")
if($message && ($action ne "Reset")){
    $times += 1;
    $message = " You have clicked me $times times. ";
} else{
    $times = 0;
    $message = "Welcome.";
}
print qq{
    <form>
        $message
        <input type=submit name=action value=Proceed>
        <input type=submit name=action value=Reset>
        <input type=hidden name=message value="$message">
        <input type=hidden name=times value="$times">
    </form>
};
```

## 2. Same with CGI/Python

```
#!usr/bin/python
import cgi
q = cgi.FieldStorage()
print "Content-type: text/html\n\n"
(times, message, action) = ("", "", "")
if q.has_key("times") : times = q["times"].value
if q.has_key("message") : message = q["message"].value
if q.has_key("action") : action = q["action"].value
if message and action != "Reset":
    times = str(int(times) + 1)
    message = " You have clicked me %s times. " % times
else:
    times = 0
    message = "Welcome."
print """
<form>
    %s
    <input type=submit name=action value=Proceed>
    <input type=submit name=action value=Reset>
    <input type=hidden name=message value="%s">
    <input type=hidden name=times value="%s">
</form>
""" % (message, message, times)
```

## 3. Write the shortest/simplest programs in PHP that keep state. (Two programs.)

First program:

```
<?
    if($message && $action != "Reset"){
        $times += 1;
        $message = "Number of clicks so far: $times";
    } else{
        $message = "Welcome.";
        $times = 0;
    }
?>
<form>
    <?=$message?>
    <input type=submit name=action value=Proceed>
    <input type=submit name=action value=Reset>
    <input type=hidden name=message value=<?=$message?>>
    <input type=hidden name=times value=<?=$times?>>
</form>
```

Second program:

```
<? session_start();
    if($message && $action != Reset"){
        $times += 1;
        $message = "Number of clicks so far: $times";
    } else{
        $message = "Welcome.";
        $times = 0;
        session_register("message");
        session_register("times");
    }
?>
<form>
    <?=$message?>
    <input type=submit name=action value=Proceed>
    <input type=submit name=action value=Reset>
</form>
```

**4. Write the simplest server-side state program that is not affected by reloads.**

```
<? session_start();
    if($message && $action != "Reset"){
        if($age == $ageCopy){
            $times += 1;
            $message = "Number of clicks so far: $times";
            $age += 1;
        } else {
            echo "Sorry, reloads are not allowed! <p>";
        }
    } else{
        $message = "Welcome.";
        $times = 0;
        $age = 1;
        session_register("message");
        session_register("times");
        session_register("age");
    }
?>
<form>
    <?=$message?>
    <input type=submit name=action value=Proceed>
    <input type=submit name=action value=Reset>
    <input type=hidden name=ageCopy value=<?=$age?>>
</form>
```

**5. Write SQL for creating a user, a database and giving the user permission to the database.**

```
create user 'lbird'@'silo.cs.indiana.edu' identified by 'drivl';
create database studyguide;
grant all on studyguide.* to 'lbird'@'silo.cs.indiana.edu';
```

We need to be root for these to work.

Getting in as lbird is similar (but not identical) to the command we use to log in as root.

**6. Write SQL to insert data in a table.**

```
mysql> insert into example
-> (session_id, message, balance)
-> values
-> ('a0000001', 'Welcome', 0);
```

**7. Write SQL for a simple query that extract some data from some table.**

```
mysql> select * from example;
```

**8. Write the simplest CGI program that keeps state server-side.**

```
#!/usr/bin/python
import cgi, random, MySQLdb, sys, os
(message, session_id, balance, action, id) = ("", "", "", "", "")
print "Content-type: text/html\n\n"
q = cgi.FieldStorage()
if q.has_key("action") : action = q["action"].value
Con=MySQLdb.Connect(host="silo.cs.indiana.edu",port=123,user="aa",passwd="qq",db="abc")
Cursor = Con.cursor()
if q.has_key("session_id") and action != "Reset":
    id = q["session_id"].value
    Cursor.execute("SELECT message, balance FROM example WHERE session_id = '%s'" %
q["session_id"].value)
    Results = Cursor.fetchall()
    (message, balance) = Results[0]
    balance = str(int(balance) + 1)
    message = " You have cliked me %s times. " % balance
    Cursor.execute("update example set message = '%s', balance = '%s' where session_id =
'%s'" % (message, balance, q["session_id"].value))
else:
    balance = 0
    message = "Welcome."
    for i in range(8): id += str(random.randrange(10))
    if not Cursor.execute("insert into example (session_id, balance, message) values ('%s', '%s',
'%s')" % (id, balance, message)):
        print "Content-type: text/html\n\nError. Please reload."
print ""
```

```

<form>
  %s <input type=submit name= action value=Proceed>
  <input type=submit name=action value=Reset>
  <input type=hidden name=session_id value=%s>
</form>
""" % (message, id)

```

**9. Write a simple program that accesses a database and extracts information. Use the language of your preference.**

```

<?
$servername = "localhost";
$username = "whoever";
$dbname = "whatever";
$password="wha55up";
$port = "26262";
$socket = "/u/duozwang/a348-spring-2021-workspace/mysql/mysql.sock";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname, $port, $socket);
if ($conn) {
    echo "I can select the database. <p>";
    $query = "select * from players"; // table of interest
    $result = $conn->query($query);
    if (! $result) echo "I don't see anything in here. <p>";
    else {
        $num_rows = $result->num_rows;
        echo "There are " . $num_rows . " records I can see. <p>";
        while ($row = $result->fetch_row()) {
            echo "(" . $row[0] . ", " . $row[1] . ") <p> ";
        }
    }
} else {
    echo "I cannot connect. <p>";
}
?>

```