**WEB SERVICES**

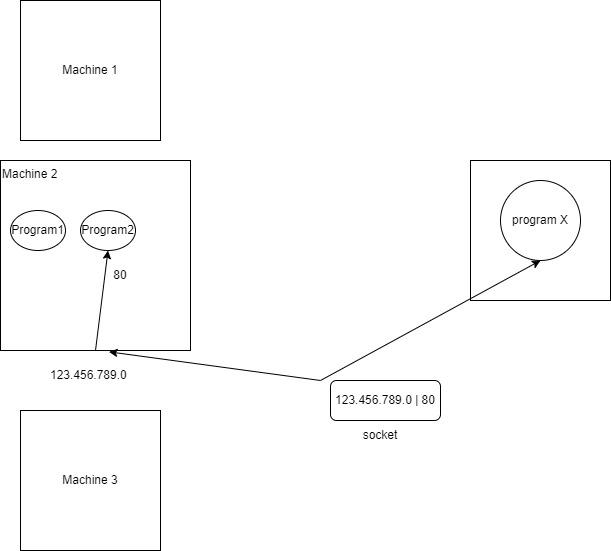
420-511-VA

#### **LAB-1-**

Note: You need to demo your completed LAB to the teacher after completion.

**How are IPs and PORTs used for communication?**

In a TCP/IP communication, an IP address is the address of a machine on a network, and a PORT number can be seen as the address of a program on the machine.   
Once data, that includes an IP address and a PORT number, is received by a machine at a certain IP address, the program that is listening to the PORT included in the data will have access to the data by the operating system and its services. For example once data arrives at a web server using it’s IP the HTTP Server (Apache) will access the data if it is addressed to PORT 80, that would be set in the TCP header of the data.



The purpose of this LAB is to understand program to program communication using sockets. This will help in understanding client server communication or more specifically web browser, the client, and web server communication in our context.

This can also be used to implement a process to process communication using sockets, or Remote Procedure Call (RPC) which consists of two processes even on two different machines communicating remotely.

We will create two programs program-1 and program-2, where program-1 sends a message, or data, to program-2 over a socket defined by the IP address of program-2 and the port number that program-2 is listening to. Upon receiving the message from program-1, program-2 will echo the message and write a feedback to program-1, program-1 will read and echo the feedback. The socket is closed at the end of the send and receive procedures.

Technically if you are working in the computer LAB on the computers you could team up with a colleague one can write program-1 on their machine and the other can write program-2 on another machine and have them communicate.

We will use the PHP Socket extension, the documentation can be found in the PHP manual:

* https://www.php.net/manual/en/book.sockets.php

Note: In similar exercises, you need to read and analyze the code, not just copy paste.

1. Create program-1

Write the below code and save it as program-1.php

Setting the target IP address and port number:

* these are arbitrary,
* we need to make sure that the IP address is accessible by the machine, we use 127.0.0.1 which corresponds to the address of the machine to itself, it is called the loopback address. It is equivalent to ‘localhost’,
* we use the port number 10000, no other program should be listening to this port otherwise a conflict occurs. We expect that there is no known program using that port number.

|  |
| --- |
| echo "sending a message to a target program, program-2, through a socket.\n";  /\* Target program IP address \*/  $ipaddress = '127.0.0.1';  /\* Target program port number \*/  $port = 10000; |

Creating the socket:

|  |
| --- |
| /\* Create a TCP/IP socket. \*/  $socket = socket\_create(AF\_INET, SOCK\_STREAM, SOL\_TCP);  echo "Socket created.\n"; |

Connecting to the socket:

|  |
| --- |
| echo "Attempting to connect to '$ipaddress' on port '$port'...";  $result = socket\_connect($socket, $ipaddress, $port);  echo "Connected to socket.\n"; |

Sending data over the socket connection:

|  |
| --- |
| /\* The data, or message, to be sent to program-2 \*/  $data = "This is a message from program-1.";  echo "Sending the data to program-2 ...";  socket\_write($socket, $data, strlen($data));  echo "Data sent.\n"; |

Reading response from the socket when program-2 sends back a message:

|  |
| --- |
| /\* Reading response from the socket \*/  echo "Reading response:\n\n";  $feedback = socket\_read($socket, 2048);    echo $feedback; |

Closing the socket:

|  |
| --- |
| echo "Closing socket...";  socket\_close($socket);  echo "Socket closed.\n\n"; |

1. Create program-2

Program-2 is similar to a service that is running, listening on a port, and waiting for data to be received. When it receives the data it will echo it and send a feedback.

Create a socket and bind it to a certain IP address and port number, these will be the same as the ones used by program-1, effectively program-1 is using these because program-2 uses them:

|  |
| --- |
| $ipaddress = '127.0.0.1';  $port = 10000;  $sock = socket\_create(AF\_INET, SOCK\_STREAM, SOL\_TCP);  /\* Bind the program to a socket, i.e., make the program listen to the specified port number \*/  socket\_bind($sock, $ipaddress, $port); |

Start listening to the socket:

|  |
| --- |
| socket\_listen($sock, 5); |

Wait for and then read the message when it comes:

|  |
| --- |
| /\* Wait for and then read the message when it comes \*/  echo "waiting for a message...\n";  $msgsock = socket\_accept($sock);  $buf = socket\_read($msgsock, 2048);  echo "Received message: $buf\n"; |

Write and send a feedback message:

|  |
| --- |
| /\* Write and send a feedback message \*/  $feedback = "Welcome to Program-2, You said '$buf'.\n";  socket\_write($msgsock, $feedback, strlen($feedback)); |

Close the socket:

|  |
| --- |
| socket\_close($msgsock);  socket\_close($sock); |

1. As explained in class, execute the programs using the command prompt (cmd), or PowerShell, to see the result.
2. Code analysis questions:
   1. Do the IP addresses and port numbers in program-1 and program-2 have to be the same? Explain why in one statement.

|  |
| --- |
| Yes, because having it the same allows the machine to send and receive its own data. |

* 1. What is the parameter ‘SOL\_TCP’ used for the function socket\_create correspond to?

|  |
| --- |
| It corresponds to the TCP protocol. |

* 1. In what situations would we use UDP as the protocol, instead of TCP? and Why?

|  |
| --- |
| UDP is used instead of TCP when it comes to broadcasting, such as streaming video or audio over the Web because guaranteed delivery is not as important as fast transmission. |

* 1. What parameters does the function socket\_write take?

|  |
| --- |
| The socket\_write function takes a (1) socket instance of the Socket class, (2) string data, and (3) the optional length parameter (alternate length of bytes written to the socket). |

* 1. What parameters does the function socket\_read take?

|  |
| --- |
| The socket\_read function takes a (1) socket instance created with socket\_create() or socket\_accept, (2) length (maximum number of bytes read), and (3) the optional mode parameter(by default is PHP\_BINARY\_READ) |