British Columbia Institute of Technology

Scalable Server Design

Assignment 2 - COMP 8005

Manuel Gonzales

A00866174

Feb 15, 2016

**THREADED SERVER DIAGRAM**



**Pseudocode**

Start Structure and Semaphores

{

Create Structure to gather data

Create Semaphores to control thread creation

Create Listening Socket

}

Spawn Threads

{

Create Threads to listen from the previously created Listening Socket

}

Accept Connections / Server Connection

{

Signal Semaphore to create new thread

After accepting a new connection read data form it and echo it back

Update data based on requests and payload

}

Close Connection

{

After Gathering all the data, save it to structure

Save it to csv file

Close Socket

}

Print Data()

{

Show current , total , peak connections

}

Stop()

{

Close all active sockets

Free memory

exit

}

**SELECT SERVER DIAGRAM**



**Pseudocode**

Start Structure and Semaphores

{

Create Structure to gather data in shared memory

Create Semaphores to control process creation

Create Listening Socket

}

Spawn Processes

{

Create processes to listen from the previously created Listening Socket

Create map for data gathering.

}

Set select

{

Set select to add new connections

Set timeout to 20 seconds

On timeout close all sockets and restart.

On error close all sockets and exit

}

Accept Connections / Server Connection

{

Let descriptor list know of new connection

Take data from connection read data form it and echo it back

Update data based on requests and payload

}

Close Connection

{

After Gathering all the data, save it to structure

Save it to csv file

Close Socket

}

Print Data()

{

Show current , total , peak connections

}

Stop()

{

Close all active sockets

Free memory

exit

}

**EPOLL SERVER DIAGRAM**



**Pseudocode**

Start Structure and Semaphores

{

Create Structure to gather data in shared memory

Create Semaphores to control process creation

Create Listening Socket

}

Spawn Processes

{

Create processes to listen from the previously created Listening Socket

Create map for data gathering.

}

Set Epoll

{

Set epoll to add new connections and wait for events

Set timeout to 20 seconds

On timeout close all sockets and restart.

On error close all sockets and exit

}

Accept Connections / Server Connection

{

Let descript list know of new connection

Take data from connection read data form it and echo it back

Update data based on requests and payload

}

Close Connection

{

After Gathering all the data, save it to structure

Save it to csv file

Close Socket

}

Print Data()

{

Show current , total , peak connections

}

Stop()

{

Close all active sockets

Free memory

exit

}

**CLIENT DIAGRAM**



Start Structure and Semaphores

{

Create Structure to gather data in shared memory

Create Semaphores to control process creation

}

Spawn Processes

{

Create processes to create threads.

Create Semaphores to control threads creation

Set structure for data gathering.

}

Spawn Threads

{

Create Threads to connect to server

}

Connect

{

Create socket and update data structure

}

Send / Receive

{

Send data based **on request number**

receive echo until all data has been received back

}

Close Connection

{

After Gathering all the data, save it to structure

Save it to csv file

Close Socket

}

Print Data()

{

Show current , total , peak connections

}

Stop()

{

Close all active sockets

Free memory

exit

}