|  |  |  |  |
| --- | --- | --- | --- |
|  | 2/12/2016 |  | |
| |  |  | | --- | --- | |  |  | |  |  | |  |  | |  |  | | |  | |
| COMP8005 Assignment 2  *Designs* | | | |
|  | Rizwan Ahmed  Vishavpreet Singh  COMP6D | |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

Contents

[Design 2](#_Toc443244907)

[Diagrams 2](#_Toc443244908)

[Multi-threaded Server 2](#_Toc443244909)

[Select server 3](#_Toc443244910)

[Epoll server 3](#_Toc443244911)

[Echo Client 4](#_Toc443244912)

[Pseudocode 5](#_Toc443244913)

[Multi-threaded Server 5](#_Toc443244914)

[Select server 5](#_Toc443244915)

# Design

## Diagrams

### Multi-threaded Server



### Select server



### Epoll server

### Echo Client



## Pseudocode

### Multi-threaded Server

Main – entry point

Set up signal handler for user pressing CTRL+C

If user presses CTRL+C: kill\_server()

Create socket

Set socket option to reuse

Set socket to listen on all addresses

Create array of threads

Start accept\_loop()

accept\_loop() – wrapper function for accept()

Use accept() call to accept connection from a client

Increase buffer size

Get client info (address, port, number of requests, data sent)

Create new thread if we need more

Have thread serve its clients

serveClients()

Get file descriptor from client

Read from file descriptor

If the client disconnected then close the file descriptor

Otherwise write to file descriptor

If thread has no more clients to write to then terminate thread

kill\_server()

Open file

Terminate all threads and connections

Write all statistics to file

Close file

Exit program

### Select server