INB/N 365 Assignment 2 Marking Guide

Student Name:

Student Number:

INB Total Mark: __/45
INN Total Mark: __/60

Report __/5
__/1 – Appropriate formatting. Statement of completeness, instructions to run the program, contents, section headings and page numbers are required otherwise score is 0.
__/1 – Description of data structures.
__/1 – Description of each program.
__/3 – Description of how processes, threads, sockets, servers and clients execute and interact including use of data structures and communication protocols.

Compilation and Execution of Programs __/5
__/1 – Programs compile.
__/1 – Programs emit no warnings during compilation.
__/1 – Programs do not hang at any stage.
__/1 – Server exits cleanly upon SIGINT (check source code).
__/1 – Programs display correct output as per the specification.

Source Code Style __/5
__/2 – All functions include a description, including descriptions of parameters and return values.
__/2 – Program is broken into appropriately sized and named functions.
__/1 – Code is clear, simple and has consistent formatting.

Use of System Resources __/5
__/2 – Program is free from buffer overflows. All writes to arrays are within array bounds for all possible executions of the program.
__/1 – All files are closed under all possible executions of the program.
__/1 – All memory is deallocated under all possible executions of the program.
__/1 – Use of malloc() is avoided where possible (i.e. for fixed sized arrays).

TASK 1: __/15
__/2 – Server takes port number as a command line parameter.
__/2 – Server reads both CRLF and LF terminated lines.
__/2 – Server parses GET request correctly.
__/2 – Server discards header lines sent by client.
__/2 – Server sends the correct response if the requested file exists.
__/2 – Server sends the correct response if the requested file does NOT exist.
__/2 – Server correctly sends file to client.
__/1 – Client is implemented as per the specification.

TASK 2: __/10
__/2 – Thread pool is limited to 30 concurrent connections
__/2 – Thread pool reuses threads and only ever creates 30 threads
__/6 – Synchronisation
– 0 marks if synchronisation is not implemented, including not using pthreads.
– 3 marks if synchronised using pthreads but contains data races or dead locks. For example, a mutex is used and the empty or full state is read without use of a condition variable or any other form of synchronisation. Any case of the shared state not being locked when being read or written can lead to a data race. C makes no guarantees about concurrent access, even if it is assumed an operation is atomic on the underlying hardware. For example, memory fences need to be used on most hardware, including x86 based hardware, to ensure writes are visible to all hardware threads. Safe concurrent access can only be achieved by the use of pthreads that implements all hardware specific synchronisation issues not specified by the C language standard.
– 6 marks if synchronised using phtreads and free from data races and dead locks.

TASK 3 (INN Students Only): __/15
__/3 – Server parses POST request correctly.
__/3 – Server parses headers sent by client for Content-Length.
__/3 – Server sends the correct output of the program if it is executable.
__/3 – Server sends the correct response if the requested file is NOT executable.
__/3 – Server correctly executes program and sends its output to the client.