

Project Update 1

Ben Soer

A00843110

Research Component

The final project for COMP7615 will be the recreation of the 10k problem in NASM assembly. The 10k problem is a typical server infrastructure problem where a servers performance can be deemed “scalable” if it is able to sustain ten thousand simultaneous connections to it. The problem has been previously implemented in COMP8005 in C/C++. The project was previously implemented also using different scaling techniques so as to benchmark and compare the different architectures. This includes a multi-threaded/multi-process architecture, an architecture using the *select* system call, and an architecture using the *epoll* system call.

For this project, recreation of the three previously stated architectures will be implemented in NASM assembly. This will also include the creation of a NASM assembly client which will then benchmark each architecture. At the end of the tests, the user will be able to view response times printed to file or the console by the client application.

Experiment Component

Implementation of the 10k problem will be using a number of new system calls in NASM assembly. These will include *sys_fork*, *sys_select*, and *sys_epoll_create* for making system calls for the primary functions of each server. Additionally the development of a client that will simulate multiple connections needs to be developed. This will simply be implemented using the *sys_fork* call, allowing multiple connections to operate simultaneously on the same client program.

On top of these system calls will be the implementation and management of socket processing. Each server will have a buffer size and need to be configured to accept and process input. For this experiment, the client and server will be sending back and forth a simple message, simulating data always in transfer. With configured parameters, the data send back and forth, port numbers and IP addresses can all be adjusted in the architecture.

The client will also need to write its contents to file, adding the usage of the *sys_write* system call. This though will add the addition of file management and organizing the file system so that it can be easily interpreted by the reader, after a simulation concludes. As these files will contain stats information about the round trip times and data quantities processed, usage of addition and division functions will be used including potentially the use of floating point numbers on NASM assembly. As

these have not been covered too far in depth, will add and additional challenge to the development of the client program.

Progress So Far

So far the multi-process server has been implemented and has been tested using a simple telnet client. There is currently a bug where parsing of console parameters is not possible and all configuration has to be hard coded within the source code before compiling. The source code to the current project can be found here: <https://github.com/bensoer/10kssembly>

In order to ensure the most ideal completion, the other server architectures using select and epoll will be implemented next. This allows for in the event that project runs short on time, that the client can be created still using C if necessary. The project would still hold some value in being able to see the server side architectures in NASM assembly.