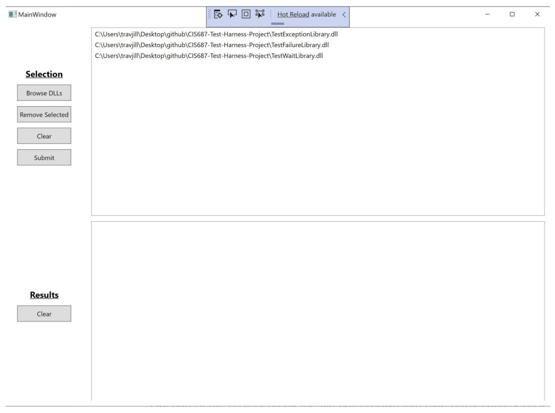
Final Group Project Submission

Team members: James Elias, Mark Hurban, Matt Reeves, Greg Shin
GitHub Repo: https://github.com/gregshin/CIS687-Test-Harness-Project

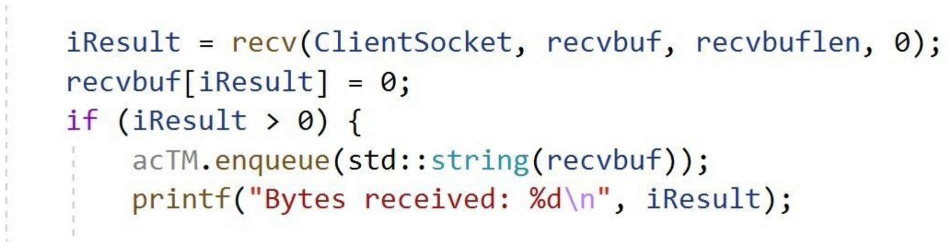
The user is able to browse for DLLs on the computer. By selecting the DLL the application readies the file path to be sent to the Test Execution via sockets. The GUI has other features including “Remove Selected” and “Clear” buttons making it easier for the user to interact with the DLLs they want executed. When the user loads all the DLLs they hit the “Submit” button and wait for the results to display on the GUI’s result section.



```
C:\Users\travjill\Desktop\github\CIS687-Test-Harness-Project\TestExceptionLibrary.dll
C:\Users\travjill\Desktop\github\CIS687-Test-Harness-Project\TestFailureLibrary.dll
C:\Users\travjill\Desktop\github\CIS687-Test-Harness-Project\TestWaitLibrary.dll
```

Figure 1: The GUI CSCI has DLL file paths loaded and is waiting the user to click the submit button

On submit the DLL file paths are sent to the Test Execution via sockets. The sockets in Gui part of the project break the file paths down into byte arrays and send it over the wire via TCP to the Test Execution CSCI. The Test Execution CSCI adds each file path to a queue and begins executing the tests in parallel using threads.



```
iResult = recv(ClientSocket, recvbuf, recvbuflen, 0);
recvbuf[iResult] = 0;
if (iResult > 0) {
    acTM.enqueue(std::string(recvbuf));
    printf("Bytes received: %d\n", iResult);
```

Figure 2: If a non empty packet is received from the GUI the Test Execution CSCI adds it to the queue

Data races are prevented using mutex whenever shared data is read/written. The test execution determines the max number of threads that can be run at once by evaluating the computer's hardware and determining the most efficient number of threads that can be run in parallel at once.

```
// Setter for maximum number of threads - sets max to total hardware can support
void ThreadManager::setMaxThreads()
{
    maxThreads = (thread::hardware_concurrency() > 2)
        ? thread::hardware_concurrency() - 1
        : 1;
};
```

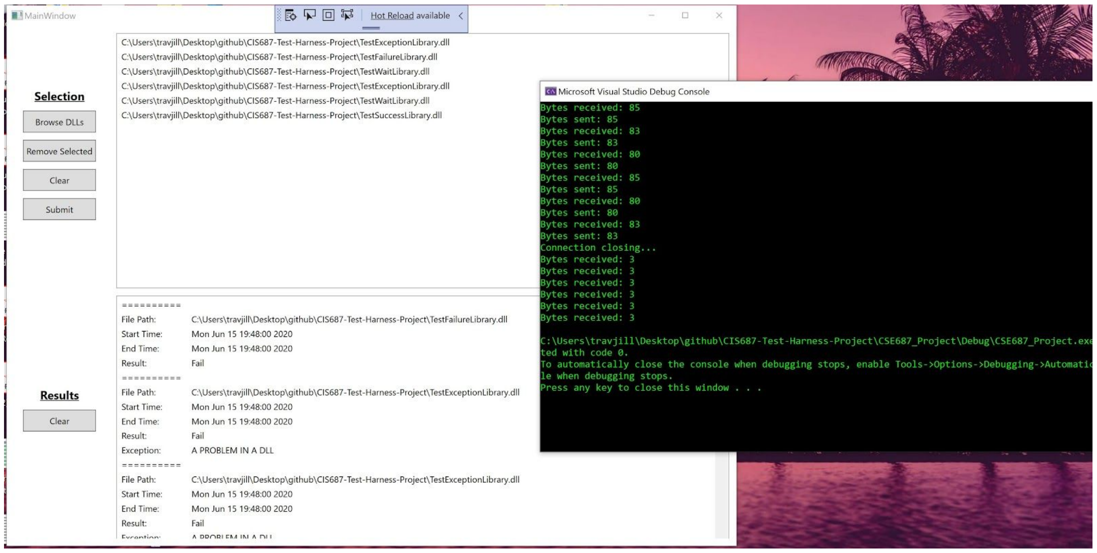
Figure 3: The Max number of threads is determined by the machine's capabilities.

If the max number of threads is being used then no more DLLs will be dequeued. Once a thread opens up another DLL is dequeued and a new thread is started, this is repeated until the queue is empty. Results are added to a shared vector (secured using mutex).

```
lock.lock();
results.push_back(dllDetails.location + dllDetails.startTime + dllDetails.endTime + dllDetails.result + dllDetails.errorMessage);
lock.unlock();
resultsAvailable.notify_one();
```

Figure 4: Whenever shared data is being written to it is locked to prevent data races.

After the tests have been completed the results are sent to the GUI using sockets. The results are broken down into byte arrays and sent over the wire using TCP. The GUI is waiting for the results and upon receiving them, displays the results to the user.



```
C:\Users\travjill\Desktop\github\CIS687-Test-Harness-Project\TestExceptionLibrary.dll
C:\Users\travjill\Desktop\github\CIS687-Test-Harness-Project\TestFailureLibrary.dll
C:\Users\travjill\Desktop\github\CIS687-Test-Harness-Project\TestWaitLibrary.dll
C:\Users\travjill\Desktop\github\CIS687-Test-Harness-Project\TestExceptionLibrary.dll
C:\Users\travjill\Desktop\github\CIS687-Test-Harness-Project\TestWaitLibrary.dll
C:\Users\travjill\Desktop\github\CIS687-Test-Harness-Project\TestSuccessLibrary.dll
```

```
==========
File Path:     C:\Users\travjill\Desktop\github\CIS687-Test-Harness-Project\TestFailureLibrary.dll
Start Time:    Mon Jun 15 19:48:00 2020
End Time:      Mon Jun 15 19:48:00 2020
Result:        Fail
==========
File Path:     C:\Users\travjill\Desktop\github\CIS687-Test-Harness-Project\TestExceptionLibrary.dll
Start Time:    Mon Jun 15 19:48:00 2020
End Time:      Mon Jun 15 19:48:00 2020
Result:        Fail
Exception:     A PROBLEM IN A DLL
==========
File Path:     C:\Users\travjill\Desktop\github\CIS687-Test-Harness-Project\TestExceptionLibrary.dll
Start Time:    Mon Jun 15 19:48:00 2020
End Time:      Mon Jun 15 19:48:00 2020
Result:        Fail
Exception:     A PROBLEM IN A DLL
```

```
Bytes received: 85
Bytes sent: 85
Bytes received: 83
Bytes sent: 83
Bytes received: 80
Bytes sent: 80
Bytes received: 85
Bytes sent: 85
Bytes received: 80
Bytes sent: 80
Bytes received: 83
Bytes sent: 83
Connection closing...
Bytes received: 3
Bytes received: 3
Bytes received: 3
Bytes received: 3
Bytes received: 3
Bytes received: 3

C:\Users\travjill\Desktop\github\CIS687-Test-Harness-Project\CSE687_Project\Debug\CSE687_Project.exe
ted with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatic
le when debugging stops.
Press any key to close this window . . .
```

Figure 5: Shows the final result once the DLLs have been submit and executed