**Project: Distributed Shared Memory**

**Jacob Ekstrand & Kamal Nadesan**

**Purpose:**

Distributed shared memory allows multiple CPUs on many machines to access the same memory. There are several advantages of using distributed shared memory that include reducing the need of constant data transfer between the machines within the system. This approach also reduces the monetary cost of building a distributed system since this system will not require as many CPUs. Our overarching project goal is to port the Password Cracker Lab to a DSM system to hold the usernames and passwords for every computer in the DSM to update and use for later tasks. That way every computer will know the password of every username that got cracked.

We will have a testing phase in which we will make sure each computer will be able to see the passwords that have been cracked in the main server which holds memory that every computer can access when they connect to the DSM.

**Project Scope:**

This project uses three of the problem solving strategies central to CSC213. The first strategy is the use of concurrency. Maintaining concurrency is critical to the success of the project since multiple processes may need to read and write to shared memory blocks. We do not expect to have to use conditional variables in our implementation because the Password Cracker Lab itself did not require them and we are simply sharing information across computers. However, we will need a mutex lock which we will use to lock the password list when we are adding a new password to out password list before sharing the list with the connected computers.

Reasonable runtime performance is also essential so that each system is able to access memory in a productive manner.

We will use Memory Management extensively in this project. Specifically, we will be using virtual memory to share the passwords with each computer. We will have a main server that each computer connects to in order to get the information. The main server will be kept in virtual memory.

We also need to use networks in order to share the information from the main server with any computer that connects to it. In order to use networks, we will likely use a similar strategy of socket to socket connection to share the information.

**Implementation Plan:**

Our first goal is to establish a simple Distributed Shared Memory system that is used to link two machines by the 29 April. This is an important step since we will need to show that we can handle concurrency issues and test our base code for correctness and robustness. It may be necessary for the Distributed Shared Memory system to incorporate a local server that will properly handle memory access requests which our test cases will also evaluate its performance and correctness.

There will be several tests for correctness and performance that will make use of testing memory accesses to a file in the DSM system that contains a dictionary. Each of these tests will be run for one computer in the system and  two computers in the system. The first test will verify that each computer can read and write into the dictionary file. The next test will force simultaneous read and write accesses, as well as write and write memory accesses between the two systems to examine if the concurrency of the system holds. We will also test the duration of the memory accesses to ensure reasonable performance. Lastly, memory use will also be tested to ensure the correct memory addresses are being accessed.

Once this goal has been completed, we will port the Password Cracker Lab to the Distributed Shared Memory system. This process should be completed by 5 May. The testing process will likely be more extensive since each machine will need to update the username/password list. This will require a test that examines the differences of the password list for the lab run on a single machine and the DSM to verify correctness and performance. We will also test the ability of the system to run the Password Cracker Lab in parallel. This goal may have potential pitfalls that we need to take into account. For example, it may be difficult to parallelize the program to run on multiple computers which could cause delays in our timeline. If we encounter this problem, we will attempt to resolve this issue by testing the Password Lab on one machine and later add the second machine. In addition, concurrency bugs may arise within our implementation, which may further cause delays.

The third goal of this project is to add n machines, where n is an integer greater than 0 and less than 100, to the Distributed Memory System. We will also use very similar testing methods as compared to the first goal. However, we will repeatedly test our base cases with one and two computers throughout the duration of this goal to ensure our new code did not break the previous code. In addition, we will expand on the pre-existing tests from the first goal to include cases where we have n connected systems.

Our fourth goal is to port the Password Lab to the Distributed Memory system that has n computers connected. The tests for this goal will mirror those for the second goal but will include cases that will test if the system can handle a much larger number of connected computers.

*First Goal:*

Our goal is to create a DSM system that allows multiple computers to access/modify a dictionary.

*Overarching Goal:*

Port Password Lab to DSM system.

Create a DSM system that holds a list of usernames and corresponding passwords from many computers that every connected computer can use.