README File

First we need to generate the trace files for each of the 4 programs using PIN tool. First setup the pin, and copy the pin tool named 'pin_tool.cpp' and all the four programs(prog1.c, prog2.c, prog3.c, prog4.c) inside the source/tools/CS622Assignment/ in the PIN directory. Compile all four programs using the instructions as mentioned. For example to compile 'prog1.c' use following command.

```
$ gcc -O3 -static -pthread prog1.c -o prog1
```

Then build the pin tool, use following command.

```
$ make obj-intel64/pin_tool.so
```

When we run the pin tool on a given program, it will generate a trace file named 'trace.out'. For example, to run the pin tool on prog1, use following command

```
$ ../../../pin -t obj-intel64/pin_tool.so -- ./prog1 8
```

Now rename 'trace.out' to 'prog1.out'. Follow the same procedure for all four programs and rename each file before running PIN tool for other programs. Then we will have four trace files corresponding to four programs named prog1.out, prog2.out, prog3.out and prog4.out. Move these files to the location where source codes for other parts of assignment are kept after unzipping.

To proceed further, first we have to divide the trace files generated to get thread-wise trace file. For this purpose file named 'thread_breaker.cpp' is present in the directory.

The 'thread_breaker.cpp' will be executed as follow.

```
 $ g++ thread_breaker.cpp
./a.out arg1
```

Where arg1 is the name of the trace file, which we want to divide. For prog1.out, we will pass 'prog1' as arg1.

After running this 'thread_breaker.cpp', 8 trace files will be generated named 'thread0.out', 'thread1.out',... 'thread7.out'.

The code for simulator is present in the file named 'simulator.cpp' and one header file named 'cache.h' is also present there. To run the simulator for any program, following steps need to be followed.

1. Generate thread-wise traces for it's trace file. For example, prog1.out will be divided like:

```
 $ g++ thread_breaker.cpp
$ ./a.out prog1
```

2. Then run the simulator.

```
 $ g++ simulator.cpp
$ ./a.out
```

Results of simulator will be printed on the console.