```
/*****************************************************************************
File:           mlsTrace tool User Manual
Authors:        Albert Myers, myers@uah.edu
                Aleksandar Milenkovic, milenkovic@computer.org
                LaCASA Laboratory
                The University of Alabama in Huntsville
Description:    This document describes the use and features of mlsTrace.
                mlsTrace is a Pin tool developed to capture and record
                memory reads and writes in multithreaded programs running under Pin.
                Each memory-referencing operation is captured and its
                descriptor is recorded into a trace file.

                IMPORTANT Note: If an instruction includes multiple memory operations,
                each operation will have a separate descriptor in the output trace.
                Intel64 instruction set includes instructions that can
                read one or two operands from memory, write the result to memory,
                or read from and write to memory. Instructions such as compare string
                can actually access a number of operands in memory.
                To avoid encoding inside a trace record we emit a trace descriptor
                for each memory access. Thus one instruction may have multiple
                trace records. Instruction set simulator should be able to replay
                the trace records correctly.

Date:           Jan 17, 2014
Modification:    REV1       : Jan 17, 2013 - instructions with 62732 version of pin tool
                 Author     : Amrish K. Tewar akt0001@uah.edu
                 REV0       : October 11, 2013 - original
                 Author     : Albert Myers, myers@uah.edu
*****************************************************************************/
```

1. Tool Setup (Compilation)

a) Requirements: Linux computer with Pin installed.
To learn more about Pin and to download the latest version visit
http://software.intel.com/en-us/articles/pin-a-dynamic-binary-instrumentation-tool.
Examples in this manual are prepared on a x86_64 quad core machine running CentOS 6.3 OS
with pin-2.12-62732.

b) Download source code files for mlsTrace:
mlsTrace.cpp
mlsTrace.h.
mlsTraceAssist.h

c) Copy source code files into Pin directory
pin-2.12-62732-gcc.4.4.7-linux/source/tools/ManualExamples

d) create obj-intel64 directory in
pin-2.12-62732-gcc.4.4.7-linux/source/tools/ManualExamples

e) Add mlsTrace to the TEST_TOOL_ROOTS line in
pin-2.12-62732-gcc.4.4.7-linux/source/tools/ManualExamples/makefile.rules

f) Add -std=c++0x to the TOOL_CXXFLAGS_NOOPT line in
pin-2.12-62732-gcc.4.4.7-linux/source/tools/Config/makefile.unix.config

g) Build tool/tools

```
<<
```

```
-s  [default 0]
        Begin emitting descriptors after executing a specified number of
        instructions
-store  [default 0]
        Trace store instructions (default)
-unique_logfile  [default 0]
        The log file names will contain the pid

Line information controls

-discard_line_info
        Discard line information for specific module. Module name should be a
        short name without path, not a symbolic link
-discard_line_info_all  [default 0]
        Discard line information for all modules.

Symbols controls

-short_name  [default 0]
        Use the shortest name for the RTN. Names with version substrings are
        preferred over the same name without the substring.
-unrestricted_rtn_size  [default 0]
        Use the unrestricted RTN size. When set the RTN size defined by the
        distance between RTN start to the beginning of next RTN.

Statistic switches

-profile  [default 0]
        print amount of memory dynamically allocated but not yet freed by the
        tool
-statistic  [default 0]
        print general statistics

General switches (available in pin and tool)

-slow_asserts  [default 0]
        Perform expensive sanity checks

>>>
```

```
<<
```

```
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
[milenka@EB245-mhealth3 ManualExamples]$ pin -t obj-intel64/mlsTrace.so -load 0 -store 1 -a -
- ls
mlsTrace: Writing to text file: mlsTrace.out2014_1_17_18.21.14.txt
mlsTrace descriptor: ThreadID,  Instruction Address, Operand Address, Operand Size, Value
mlsTrace: thread begin 0 8386
. . .

[milenka@EB245-mhealth3 ManualExamples]$ head mlsTrace.out2014_1_17_18.21.14.txt
0, 0x0000003f83200b03, 0x00007fff7d508d18, 8, 0x0000003f83200b08
0, 0x0000003f83201130, 0x00007fff7d508d10, 8, 0x0000000000000000
0, 0x0000003f83201134, 0x00007fff7d508d08, 8, 0x0000000000000000
0, 0x0000003f83201136, 0x00007fff7d508d00, 8, 0x0000000000000000
0, 0x0000003f83201138, 0x00007fff7d508cf8, 8, 0x0000000000000000
0, 0x0000003f8320113a, 0x00007fff7d508cf0, 8, 0x0000000000000000
0, 0x0000003f8320113c, 0x00007fff7d508ce8, 8, 0x0000000000000000
0, 0x0000003f8320115d, 0x0000003f8341fd48, 8, 0x000c6fe9eee7b7df
0, 0x0000003f8320116e, 0x0000003f83420998, 8, 0x0000000000000000
0, 0x0000003f83201178, 0x0000003f834209a8, 8, 0x0000003f8341fdf0
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

e) Example run to trace /bin/ls shell command.
In this example mlsTrace captures both the load and the store value trace of ls (list) shell command.
The entire trace is captured in a text file. Note a change in the trace record format: an additional field indicates a read (L) or write (S) operation.

```
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
[milenka@EB245-mhealth3 ManualExamples]$ pin -t obj-intel64/mlsTrace.so -load 1 -store 1 -a -
- ls
mlsTrace: Writing to text file: mlsTrace.out2014_1_17_18.29.59.txt
mlsTrace descriptor: ThreadID, Load/Store,  Instruction Address, Operand Address, Operand
Size, Value
mlsTrace: thread begin 0 8465
. . .

[milenka@EB245-mhealth3 ManualExamples]$ head mlsTrace.out2014_1_17_18.29.59.txt
0, S, 0x0000003f83200b03, 0x00007fffeef882d8, 8, 0x0000003f83200b08
0, S, 0x0000003f83201130, 0x00007fffeef882d0, 8, 0x0000000000000000
0, S, 0x0000003f83201134, 0x00007fffeef882c8, 8, 0x0000000000000000
0, S, 0x0000003f83201136, 0x00007fffeef882c0, 8, 0x0000000000000000
0, S, 0x0000003f83201138, 0x00007fffeef882b8, 8, 0x0000000000000000
0, S, 0x0000003f8320113a, 0x00007fffeef882b0, 8, 0x0000000000000000
0, S, 0x0000003f8320113c, 0x00007fffeef882a8, 8, 0x0000000000000000
0, L, 0x0000003f83201156, 0x0000003f8341fb80, 8, 0x0000003f83201130
0, S, 0x0000003f8320115d, 0x0000003f8341fd48, 8, 0x000c71887fb3be8a
0, L, 0x0000003f83201167, 0x0000003f8341ffc8, 8, 0x0000003f8341fdf0
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

f) Tracing a multithreaded program.
Use our Matrix_Multiplication_OpenMP.cpp as a text program as follows.

```
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
[milenka@EB245-mhealth3 ManualExamples]$ export OMP_NUM_THREADS=8

[milenka@EB245-mhealth3 ManualExamples]$ pin -t obj-intel64/mlsTrace.so -load 1 -store 1 -a -
- ./Matrix_Multiplication_OpenMP 16
mlsTrace: Writing to text file: mlsTrace.out2014_1_17_19.44.59.txt
mlsTrace descriptor: ThreadID, Load/Store,  Instruction Address, Operand Address, Operand
Size, Value
mlsTrace: thread begin 0 9130
```

```
mlsTrace: thread begin 1 9138
mlsTrace: thread begin 2 9139
mlsTrace: thread begin 3 9140
mlsTrace: thread begin 4 9141
mlsTrace: thread begin 5 9142
mlsTrace: thread begin 6 9143
mlsTrace: thread begin 7 9144
185
[milenka@EB245-mhealth3 ManualExamples]$ cat mlsTrace.out2014_1_17_19.44.59.Statistics
Instrumentation Time: 3697.81 ms
Number of Threads: 8
Traced 11853320 instructions
Skipped 0 instructions
Total Load Operands: 1885047
        126290 ( %6.70 ) Byte Operands
        8510 ( %0.45 ) Word Operands
        1492836 ( %79.19 ) Doubleword Operands
        256557 ( %13.61 ) Quadword Operands
        0 ( %0.00 ) Extended Precision Operands
        854 ( %0.05 ) Octaword Operands
        0 ( %0.00 ) Hexaword Operands
        0 ( %0.00 ) Operands of other size
Total Store Operands: 203794
        7171 ( %3.52 ) Byte Operands
        104 ( %0.05 ) Word Operands
        41298 ( %20.26 ) Doubleword Operands
        154830 ( %75.97 ) Quadword Operands
        0 ( %0.00 ) Extended Precision Operands
        391 ( %0.19 ) Octaword Operands
        0 ( %0.00 ) Hexaword Operands
        0 ( %0.00 ) Operands of other size

[milenka@EB245-mhealth3 ManualExamples]$ head mlsTrace.out2014_1_17_19.44.59.txt
0, S, 0x0000003f83200b03, 0x00007fffbab90b88, 8, 0x0000003f83200b08
0, S, 0x0000003f83201130, 0x00007fffbab90b80, 8, 0x0000000000000000
0, S, 0x0000003f83201134, 0x00007fffbab90b78, 8, 0x0000000000000000
0, S, 0x0000003f83201136, 0x00007fffbab90b70, 8, 0x0000000000000000
0, S, 0x0000003f83201138, 0x00007fffbab90b68, 8, 0x0000000000000000
0, S, 0x0000003f8320113a, 0x00007fffbab90b60, 8, 0x0000000000000000
0, S, 0x0000003f8320113c, 0x00007fffbab90b58, 8, 0x0000000000000000
0, L, 0x0000003f83201156, 0x0000003f8341fb80, 8, 0x0000003f83201130
0, S, 0x0000003f8320115d, 0x0000003f8341fd48, 8, 0x000c7f6a171f8442
0, L, 0x0000003f83201167, 0x0000003f8341ffc8, 8, 0x0000003f8341fdf0
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```