**Homework 4**

**Mohammad Shughri 11051096**

**Results using the factorial of 6**

**Recursive factorial results:**

Task2

Total size of the memory operations: 10619

The ratio of the read operations is: 0.571711

Task3

Frequency Of RAX 2211

Frequency Of RBX 610

Frequency Of RCX 655

Frequency Of RDX 1044

Frequency Of RSI 654

Frequency Of RDI 915

Frequency Of RSP 3450

Frequency Of RBP 407

Frequency Of R8 115

Frequency Of R9 115

Frequency Of R10 210

Frequency Of R11 58

Frequency Of R12 324

Frequency Of R13 251

Frequency Of R14 202

Frequency Of R15 195

**Iterative factorial results:**

Task2

Total size of the memory operations: 10619

The ratio of the read operations is: 0.571711

Task3

Frequency Of RAX 2211

Frequency Of RBX 612

Frequency Of RCX 655

Frequency Of RDX 1044

Frequency Of RSI 654

Frequency Of RDI 915

Frequency Of RSP 3454

Frequency Of RBP 408

Frequency Of R8 115

Frequency Of R9 115

Frequency Of R10 210

Frequency Of R11 58

Frequency Of R12 324

Frequency Of R13 251

Frequency Of R14 202

Frequency Of R15 195

**Note:** In recursive and iterative factorial I got the same results for memory operations and the frequencies using the provided factorial code.

**Steps**

After downloading the PIN tool, we have to apply the following command in order to provide Ubuntu with the path of the PIN tool.

set INTEL\_JIT\_PROFILER /home/mohammad/Downloads/pin-3.17-98314-g0c048d619-gcc-linux/intel64/lib/libpinjitprofiling.so

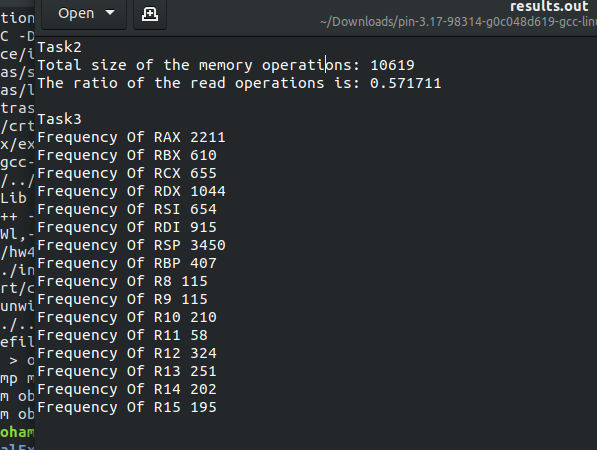
We change our directory to the tools file inside the PIN tool.

Cd /Downloads/pin-3.17-98314-g0c048d619-gcc-linux/source/tools/ManualExamples$

Inside the ManualExample, we put in our c++ code (sy.cpp) then, we create a folder obj-intel64 and put the factorial code inside it. Then, we run the c code using (gcc RecFact.c) then we execute the code using (./a.out). We put in a number and we get a new file (a.out). We move the file from obj-intel64 to the ManualExample file. After that, we will use the following command to execute our C++ code in order to count the instructions. (make sy.test).

Moreover, we will then execute our instruction c++ code on the a.out file that we got from the factorial using the command: (../../../pin -t obj-intel64/sy.so -- ./a.out)

Using this command will perform the count of the instructions and then it will give us a file that has the results of our calculations called results.out.



**Preview of the results.out file.**

**First task**

Not done.

**Second task**

1. We start by initializing new variables that will store the values of the memory operations and the read operations.

static UINT32 CountmemOP = 0;

static UINT32 CountReadOp = 0;

1. Using the method provided in the PIN manual, we created a for loop that will look for the memory operations and add them to our variables CountmemOP and CountReadOp.

UINT32 memOperands = INS\_MemoryOperandCount(ins);

// Iterate over each memory operand of the instruction.

for (UINT32 memOp = 0; memOp < memOperands; memOp++){

CountmemOP++;

if (INS\_MemoryOperandIsRead(ins, memOp)){

CountReadOp++;

}

}

1. Finally, we print out the results that we got for the memory operations and the read operations.

OutFile << "Task2 " << endl;

OutFile << "Total size of the memory operations: " << CountmemOP << endl;

OutFile << "The ratio of the read operations is: " << (double)CountReadOp/(double)CountmemOP << endl;

**Third task**

Each time we get one of the registers we calculate the frequencies by adding each time it occurs using if statements.

OutFile << "Task3 " << endl;

OutFile << "Freqquency Of RAX " << S << endl;

OutFile << "Freqquency Of RBX " << S2 << endl;

OutFile << "Freqquency Of RCX " << S3 << endl;

OutFile << "Freqquency Of RDX " << S4 << endl;

OutFile << "Freqquency Of RSI " << S5 << endl;

OutFile << "Freqquency Of RDI " << S6 << endl;

OutFile << "Freqquency Of RSP " << S7 << endl;

OutFile << "Freqquency Of RBP " << S8 << endl;

OutFile << "Freqquency Of R8 " << S9 << endl;

OutFile << "Freqquency Of R9 " << S10 << endl;

OutFile << "Freqquency Of R10 " << S11 << endl;

OutFile << "Freqquency Of R11 " << S12 << endl;

OutFile << "Freqquency Of R12 " << S13 << endl;

OutFile << "Freqquency Of R13 " << S14 << endl;

OutFile << "Freqquency Of R14 " << S15 << endl;

OutFile << "Freqquency Of R15 " << S16 << endl;