

Note1: All comments of the codes are written in the cpp files.

Note2: Before start, please run below commands.

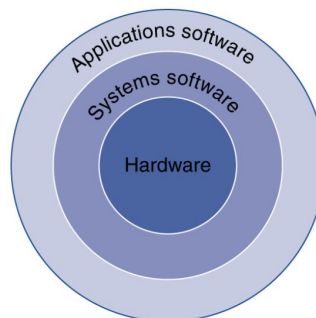
```
# Run matmul application.
>>> chmod +x matmul.sh
>>> ./matmul.sh -o a.out
# Enter folder to count instructions.
>>> cd source/tools/ManualExamples
>>> make all TARGET=intel64
# Run ./pin and write to note.txt file.
>>> ../../../../pin -t obj-intel64/inscount0.so -o note.txt -- ./a.out
```

1. Task

| | |
|--------|--------|
| Count: | 849446 |
|--------|--------|

Note: This program counts the macro instructions, since these instructions are generally in assembly language and also Intel PIN tool find the assembly level codes. However when we look at the macro instructions, they are in more deep level which is hardware level.

- Systems Software (Micro Instructions -> Macro Instructions)



2. Task

```
# Make again to run inscount0.cpp file.
>>> make
```

| | |
|---------------|--------|
| Count Branch: | 75802 |
| Count Memory: | 344554 |
| Count Others: | 429090 |

3. Task

Note: I just controlled registers which wrote a value by RegWContain, since we need to find destination registers. For example,

- **add r1, r2, r3**

In this case, r1 will be our destination register. That's why, I counted register as a destination register by using RegWContain.

```
# Make again to run inscount0.cpp file.  
>>> make
```

| Registers | Counters |
|-----------|----------|
| RAX | 1580 |
| RBX | 387 |
| RCX | 477 |
| RDX | 774 |
| RSI | 466 |
| RDI | 592 |
| RSP | 1995 |
| RBP | 144 |
| R8 | 96 |
| R9 | 104 |
| R10 | 142 |
| R11 | 35 |
| R12 | 158 |
| R13 | 111 |
| R14 | 105 |
| R15 | 105 |