Course: ECE-T580 Compiler Design
Date: 09/23/2021
Instructor: Dr. Kandasamy
Teaching Assistant: Mr. Shihao Song
Ms. Arghavan Mohammadhassani

Group Student: Quoc Thinh Vo - 14419634
Tuong Tran – 14190713

Project: LLVM Tutorial

I. Project implementation and output:

The function getInstructionCount() was implemented to get the number of lines of instructions.

Code snippet:

```
for (Module::const_iterator i = m.get()->getFunctionList().begin(),
     e = m.get()->getFunctionList().end(); i != e; ++i)
{
    if (!i->isDeclaration())
    {
        outs() << "Function name: " << i->getName() << "\n";
        outs() << "Number of lines: " << i->getInstructionCount() << "\n";
    }
}

return 0;
```

Figure 1: Functionality code snippet

The output is captured as following:

```
llvm@llvm:~/ECET480/tutorial/llvm_inter_tools$ cd ..
llvm@llvm:~/ECET480/tutorial$ cd first_llvm_project/
llvm@llvm:~/ECET480/tutorial/first_llvm_project$ ls
hello  hello.cc  makefile  sum.linked.bc
llvm@llvm:~/ECET480/tutorial/first_llvm_project$ ./hello sum.linked.bc
Function name: main
Number of lines: 8
Function name: sum
Number of lines: 8
llvm@llvm:~/ECET480/tutorial/first_llvm_project$
```

Figure 2: Program output

After iterating through the function list, the two functions were “main” and “sum” and the number of lines of instructions in each function were both 8.

II. Verify the correctness of the output:

By inspecting the two files: main.ll and sum.ll, it can be easily inspected that each function has 8 lines of instructions inside its function definition.

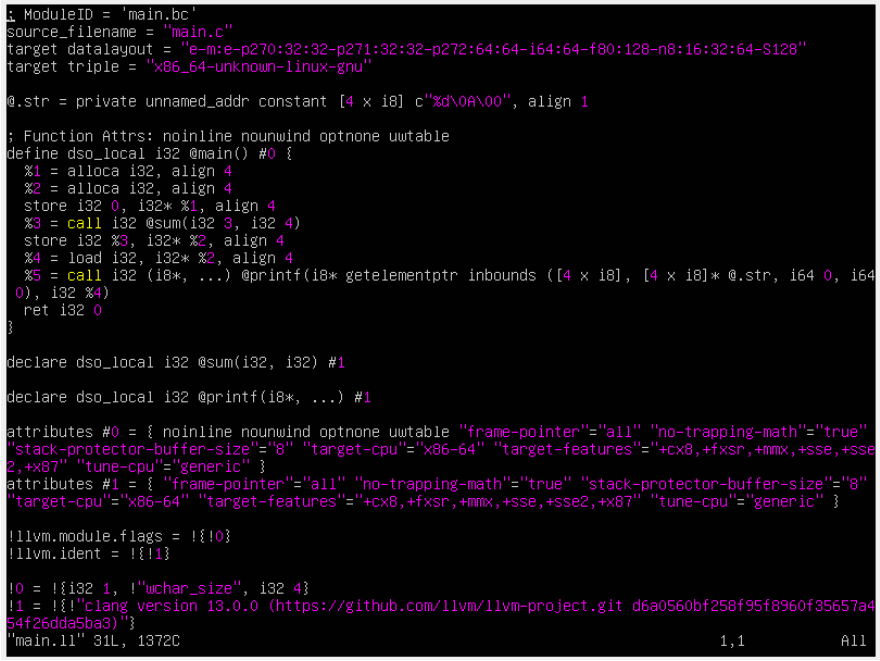

Figure 3: Function main() code inspection

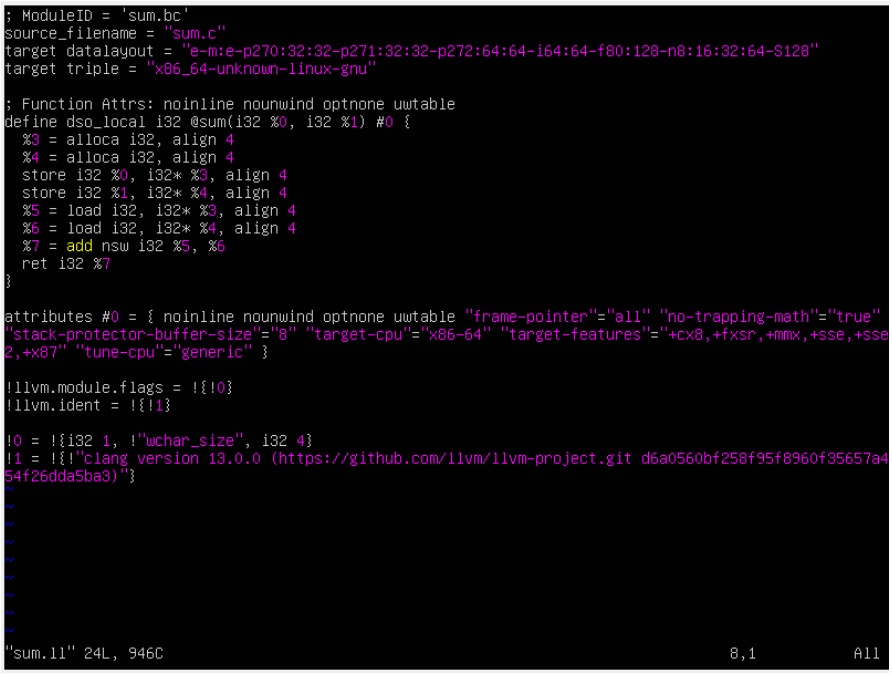

Figure 4: Function sum() code inspection

Therefore, the program output was correct with both functions' name and lines of instructions.