**UNIVERSIDADE FEDERAL DE UBERLÂNDIA**

**Trabalho de Construção de Compiladores**

**Estruturas de Representação Intermediária (IR)**

**Aluno:** Marcelo Mendonça Borges

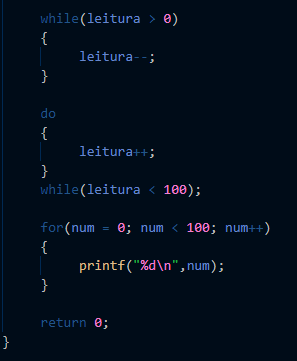
**Matrícula:** 11611BCC020

**2019**

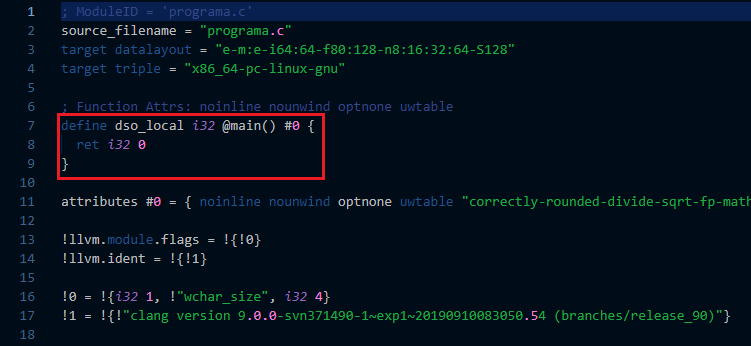
**Mapeamento de Instruções**

Foi utilizado um código em linguagem C para identificar que tipo de alteração é gerada na representação intermediária do Clang do LLVM. Esse código foi iniciado com duas bibliotecas inclusas (stdio.h e stdlib.h), e a função main vazia. Partindo disso foi incrementado as seguintes estruturas básicas da linguagem:

1. Retorno da função main;
2. Declaração de variáveis;
3. Atribuição de Valores;
4. Função printf();
5. Função scanf();
6. Estrutura if-else;
7. Estrutura while;
8. Estrutura do-while;
9. Estrutura for;



A seguir será indicado cada alteração feita no código fonte e sua alteração correspondente na representação intermediária:

**0) Situação Inicial (Com função main destacada)**

**1) Retorno da main**

****

**2) Declaração de Variável**

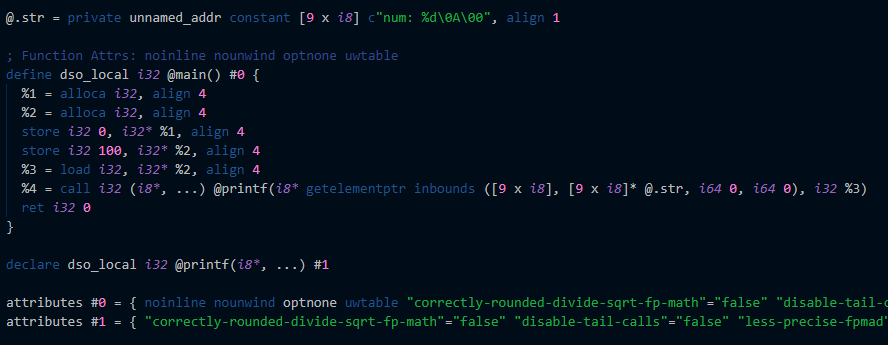
****

**3) Atribuição de Valor**

****

**4) Função printf()**

No caso da função printf(), antes da estrutura da main é adicionado o valor de @.str, além da alteração dentro da main, com a chamada da função em %4.

****

**5) Função scanf()**

No caso da função scanf(), a mesma estrutura de printf() é adicionada assim como a chamada de função na main.

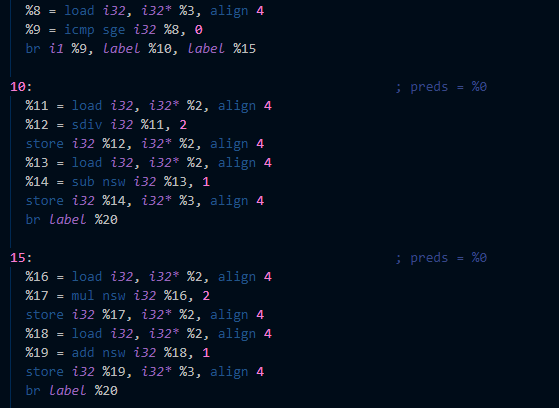


****

**6) Estrutura if-else (E operações aritméticas)**

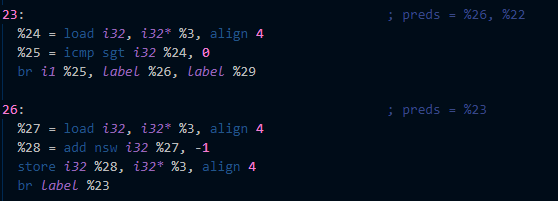
No caso dessa estrutura tem-se a parte que verifica a condição, e logo depois os dois trechos de códigos representando o código dentro de if e else, respectivamente. Nota-se também que dentro de cada um desses blocos tem as operações de divisão, subtração, multiplicação e soma, junto com atribuições.

As operações de divisão e subtração são indicadas por sdiv (%12) e sub (%10), dentro do bloco de if (10). As operações de multiplicação e soma são indicadas por mul (%17) e add (%19), dentro do bloco de else (15).



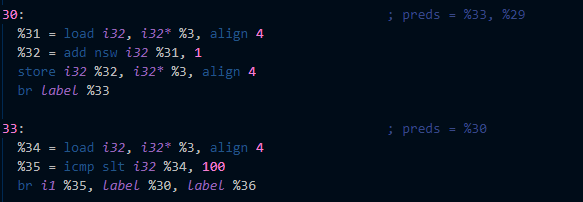
**7) Estrutura while (Com decremento)**

Nessa estrutura tem-se também o bloco de verificação da condição (23) e logo depois o bloco executável do while (26). Percebe-se também o decremento (%28).



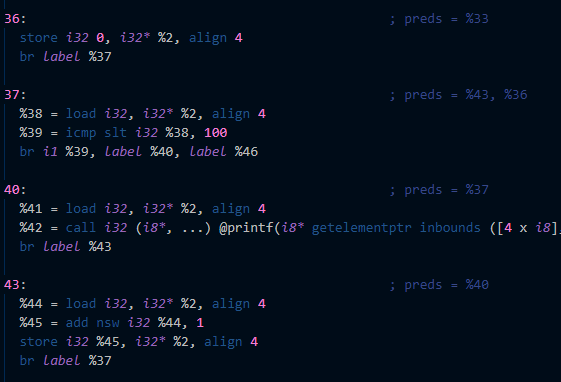
**8) Estrutura do-while (Com incremento)**

Nessa estrutura também tem o bloco de verificação da condição e o bloco executável, entretanto o executável aparece antes da verificação, como já é esperado. Temos então o executável (30) e a verificação (33). Percebe-se também o incremento (%32)



**9) Estrutura for (Com printf())**

Nessa estrutura tem a verificação da condição no início (36 e 37), o bloco de execução (40) em que pode-se perceber a função printf() (%42), e por fim temos o incremento do for (43).

****