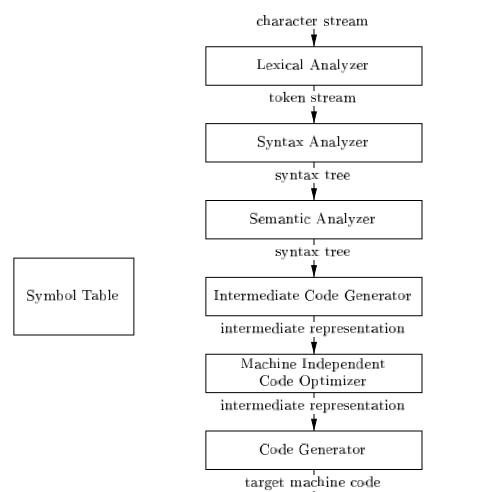
**Report of CSE 504: Assignment 1**

**Implementing the OpenMP Allocate Directive**

1. Implementation of Flush Directive:

1). During the implementation of Flush Directive, it is mainly go through the general working flow of a compiler, that is Lexical Analyze🡪Syntax Analyze🡪Sematic Analyze🡪IR Generate🡪Code Generate, details are:



2). When Implement the Flush directive, it first defined the related data structure and insert into AST, and generate OpenMPRuntime Library call to generate from AST to IR. The corresponding AST code and IR are as below:

AST:



**IR:**



3). 'omp flush' directive does not have clauses but have an optional list of variables to flush. This list of variables is stored within some fake clause, that is Flush Clause. This clause does not exist by itself, it can be only as a part of 'omp flush' directive. In other words, Flush directive use Flush Clause to store its parameter—a Variables’ list.

1. Implementation of Allocate Directive

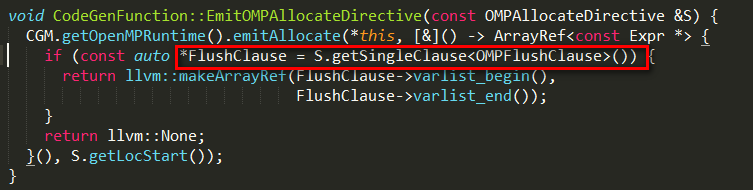
The implementation of Allocate Directive is similar with Flush Directive, but we need to deal with three problems:

a. If we use the similar code structure, do we need to new a corresponding Clause?

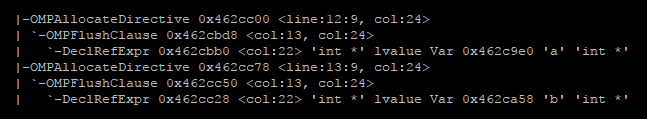
b. How to call malloc function Internally?

c. how to store the original pointer back to Allocate directive?

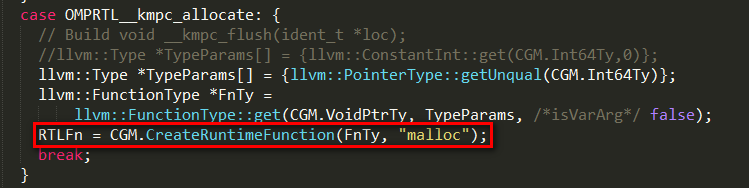
1. For Q1, the code of Allocate Directive part is the same as flush directive, just construct the code follow the Flush directive. As to Clause, coz we don’t really need to use the clause, so we can use the Flush Clause instead. For Example:



And the generated AST is:

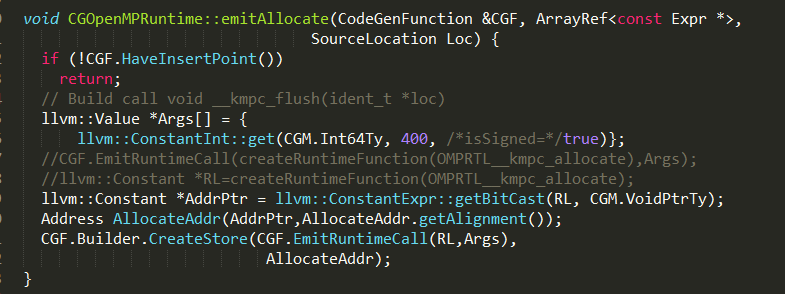


1. For Q2, in order to make a call of internal malloc function, we need to implement codes in CGOpenMPRuntime.cpp as below, the statement in red will create runtime function to use internal malloc and return the address:



1. For Q3, what we need to do is to store the original pointer back to Allocate function use the CreateStore(\*value,Address), but I am a little lost in the pointers, but I got the below IR.

**EmitRuntimeCall and CreateStore:**



**IR**:

