**课程编号：A0801041040**

**编译方法结课作业**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **姓名** | **柳成林** | **学号** | | **20206722** |
| **班级** | **软件2001** | **任课教师** | | **刘洪娟** |
| **开设学期** | **2021-2022秋季学期** | | | |
| **评定成绩** |  | | **评定人** | **刘洪娟** |
| **评定日期** | **2022年11月1日** |

**东北大学软件学院**

题目1：编译程序中使用的关键技术都有哪些应用方向，请详细说明（本题答案不得少于1页）。

**1、反编译技术：**

反编译是将可执行的（准备运行的）程序代码（也称为目标代码）转换为某种形式的高级编程语言，使其具有更易读的格式。反编译是一种逆向工程，它的作用与编译器的作用相反。它与编译相反。完成此任务的工具称为反编译器。反编译有许多不同的原因，例如理解程序、恢复源代码以进行存档或更新、查找病毒、调试程序和翻译过时的代码。完全自动化的反编译是不可能的。没有反编译器可以获得开发人员编写的确切源代码。

**2、编译技术在反病毒方面的应用**

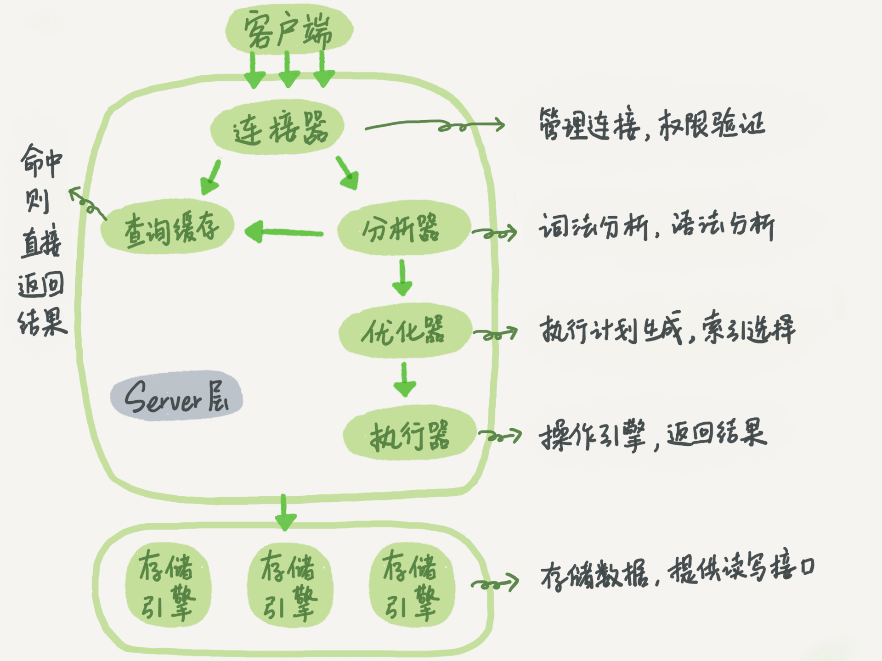
被病毒感染的网页类文件经过词法分析和语法分析，就可得到一有限自动机软件，即虚拟机，此种方式的查病毒虚拟机不同于以前虚拟机查毒方式，以前查杀病毒时的环境就像一个高级机械手，打开、定位、查杀，而此种方式则是类似一个智能探测器，打开、分析、查杀。在词法规则分析时“空格、注释”就可以被置不理了， 有和没有并不影响定位病毒代码的位置，病毒代码在同样的位置可被找到。

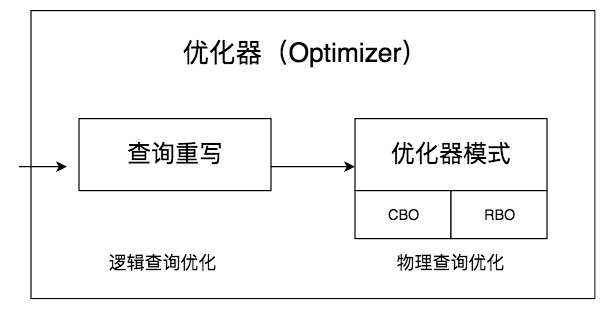
**3、基于编译技术的协议解析方法**

在过程工业的控制中存在着大量的通信协议，这些协议的结构差别很大。 要进行上层应用开发，必须对这些协议进行解析和处理。基于编译技术的协议解析方法用形式化描述的方法对协议进行描述，实现了与协议无关的协议解析和处理，从而避免了针对不同通信协议均要编写相应的解析和处理程序，使协议的解析和处理具有更好的灵活性和普适性。虽然，极少有人会去构造一个编译程序，但是编译课程中所介绍的一些原理、方法和算法并不局限于编译。 可以说，不仅仅是使用计算机编写程序，甚至仅仅使用应用软件，那么编译技术就如守护进程一样一直在幕后支持和陪伴着。所以说，计算机系统的方方面面都在运用和应用着编译领域的技术和成果。

**4、数据库查询优化器：**

SQL语言也是一种编译型语言，需要SQL编译器编译后才能执行，但它与C、C++、Java等语言不同，SQL语言是一种非过程化语言，这意味着使用SQL进行操作的时候，你只需要指定你要达到什么目的，而无需指明要怎样达到目的。解决“怎么做”问题的工具就是“优化器”。





优化器中也使用了语法分析（Parse Query）和语义检查等编译程序的技术，SQL语句被提交给SQL编译器，编译器分析该语句，进行语法分析，如果存在语法错误，编译器就停止处理并返回错误信息；如果不存在语法问题，编译器会将SQL语句转换为可被优化器分析的关系代数语句，并据此创建该查询的查询图模型（Query Graph Model，又称语法树）。语法分析完成后，编译器会根据查询图模型进行语义检查（比如检查语句中的数据类型是否与数据库的表列的数据类型一致），语义检查完成后也会将相关信息添加到查询图模型，包括参考约束，表检查约束，触发器，和视图信息等。

题目2：基于中间代码的优化中，循环的查找算法有哪些？循环优化的方法有哪些？举例说明（本题答案不得少于1页）。

**循环的查找算法：**

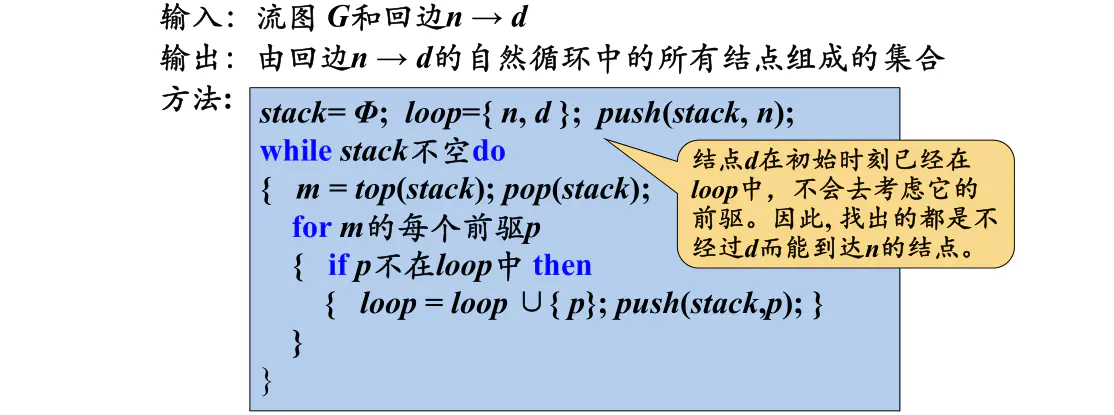
**1）利用控制点（dominator）查找循环**

首先要在中间代码上划分出基本块并且构造出控制流图（CFG）。控制流图是个可能有环的有向图。要找出其中的循环，最经典的做法就是使用Tarjan强联通算法找出所有强连通分量，如果一个强连通分量是多于一个控制流图节点（基本块）的那么肯定有循环，然后根据dominator关系确定loop header是谁；如果一个强联通是单一节点（基本块）的，则要检查该节点是否有从自己的末尾到自己的开头的控制流边，如有则是循环。这样就可以找出所有循环了。接下来通常会做的事情是把识别出来的循环归纳为一棵“循环树”（loop tree）或者说循环森林（多棵循环树），以描述循环之间的嵌套关系。此处的Tarjan强联通算法可以换为DFS。

**2）自然循环：利用回边查找循环**

在自然循环：有唯一的入口结点，是循环的唯一入口；循环中至少有一条返回入口结点的回路。自然循环是适合于优化的一种循环，如果存在回边使其能够完成自然循环，则存在自然循环。

自然循环识别算法步骤：假设n->d是回边，则该回边构成的循环包含下列结点n、d以及不经过d能到达n的所有结点。为了找到自然循环，首先要把n->d加入循环L中，如果n!=d且n的父节点不在L中，则将它加入L中L := L∪{n’}，对上述父节点重复求解，直到没有新节点加入，算法结束，得出的L中的点即为一个循环。



**3）强连通分量算法**

图的连通性通常与循环有着潜在的联系，对于无向图，只要形成回路即存在循环，但有向图不是。在有向图G中，如果两个顶点间V\_i，V\_j有一条从V\_i到V\_j的有向路径，且存在一条从V\_j到V\_i的有向路径，则两个顶点强连通，如果有向图G中每两个顶点都强连通，则G是一个强连通图，有向图的极大强连通子图，称为强连通分量（SCC）。

如果图中的SCC中存在两个结点及以上，即说明存在循环。对强连通分量进行求解并判断，也是循环的一种查找方法，强连通分量的求解算法共有三种：Kosaraju算法、Tarjan算法、Garbow算法。

**4）深度优先遍历(Depth first traverse)算法**

深度优先遍历算法（DFS）：从图的某一结点出发，访问此结点，然后依次对未访问过的邻接点进行访问，如果邻接结点已被访问或已访问完毕，则返回上一个结点，循环往复，直至图中所有结点都被访问为止。如果在遍历过程中发现邻接结点已被访问，则说明图中存在回路。在写程序代码中，时常出现函数嵌套，无明显循环出现，递归遍历的深度优先遍历可以找出程序中潜在的循环，比较直观。

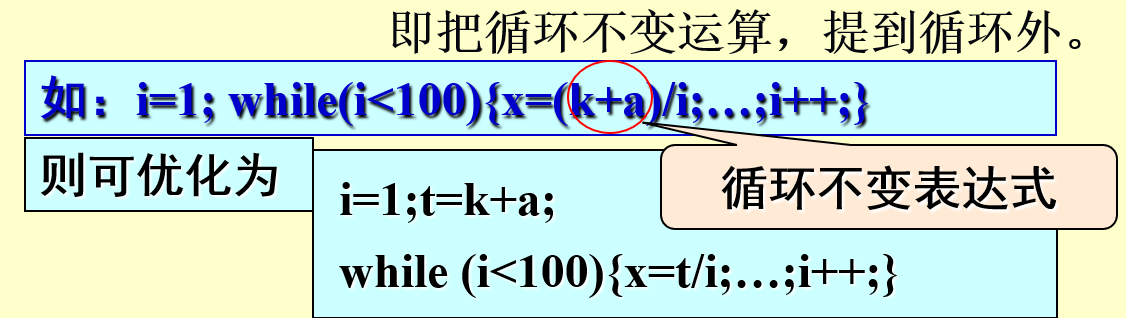
**循环优化算法**

**1、不变表达式外提**

循环中的某些代码虽然随着循环反复地执行，但其中某些运算的结果并没有发生改变，例如循环中有形如A:=B op C的代码，并且如果B和C都是常数，或者到达它们的定值点都在循环外，那么不管循环多少次，每次计算出来得到的B op C的结果都是不变的，对于这样的不变运算，我们完全可以将其外提到循环以外，避免其随着循环多次计算。程序的结果没有发生变化，但程序的运行速率却得到了一定程度的提高，这就是代码外提。

实行代码外提时，我们在循环的入口结点之前建立一个新结点(基本块)，称为循环的前置结点，该前置结点以循环的入口结点为其唯一后继。并且原来流图中从循环外引到循环入口结点的有向边，都改为引到该前置结点。

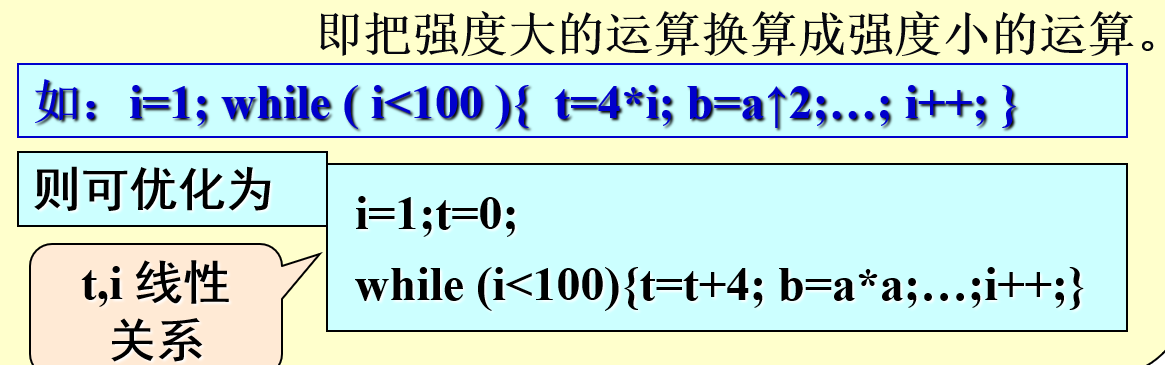
在以下的例子（来自ppt）中，k+a的结果在每一次循环中都相同，可以将其外提，讲k+a赋值给t，减少运算次数。



**2、消减运算强度**

强度削弱是将程序中执行时间较长的运算替换为执行时间较短的运算。例如，最常见的就是将循环中的乘法运算用递归的加法运算来代替，因为加法比乘法快。这样可以降低计算复杂度。

在以下的例子（来自ppt）中，将乘法换成加法运算。



**3、删除归纳变量**

在循环中如果对于变量A的赋值只有唯一的、形如A:=A±b1的赋值，那么称A为循环中的基本归纳变量,而当A是一个基本归纳变量，循环中对于B的赋值总是可以化归为A的同一个线性函数，即B:=k\*A±b2，则B是一个归纳变量。删除归纳变量的具体算法是：

1.利用循环不变运算的信息，找出所有基本归纳变量;

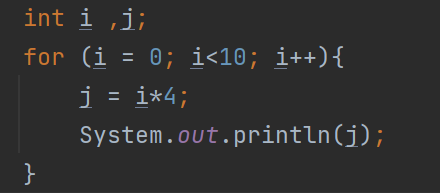
2．找出归纳变量A，并且找出A与基本归纳变量的线性函数关系A=kx + b;

3．对于每一个归纳变量A进行强度削弱;

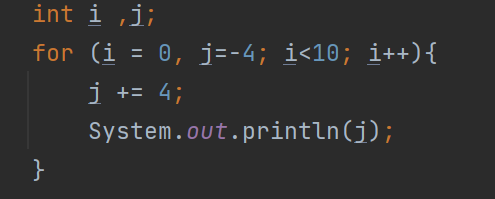
4．删除对归纳变量的无用赋值;

5．删除基本归纳变量。如果基本归纳变量在循环中除了用于自身的递归赋值以外，只在形如:if x rop c goto L的语句中被引用，则可以选取与x同族的归纳变量A来代替x进行循环控制，而后删除循环中对x赋值的代码

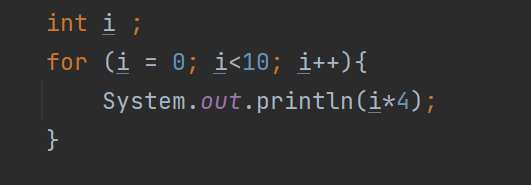
例如以下的一段循环代码：



对j进行强度消减

****

消除归纳变量j

****

**成绩评价表格**：

|  |  |
| --- | --- |
| 考核标准 | 得分 |
| （1）正确理解和掌握编译程序的工作原理；（30%）； |  |
| （2）有自己独到的见解（20%）； |  |
| （3）文字描述能够使用专业术语完成（30%）； |  |
| （4）论述有理论依据（20%） |  |