问题背景：commit16 中关于 parse() 语法分析部分

首先是语法规则的定义

```
// program = "{" compoundStmt
// compoundStmt = stmt* "}"
// stmt = "return" expr ";"
//        | "if" "(" expr ")" stmt ("else" stmt)?
//        | "for" "(" exprStmt expr? ";" expr? ")" stmt
//        | "{" compoundStmt
//        | exprStmt
// exprStmt = expr? ";"
// expr = assign
// assign = equality ("=" assign)?
// equality = relational ("==" relational | "!=" relational)*
// relational = add ("<" add | "<=" add | ">" add | ">=" add)*
// add = mul ("+" mul | "-" mul)*
// mul = unary ("*" unary | "/" unary)*
// unary = ("+" | "-") unary | primary
// primary = "(" expr ")" | ident | num
```

可以看到在 stmt 中处理了 for 循环语句

Q：为什么 for() 循环中使用 exprStmt 和 expr 去处理

因为 `for(init ; condition ; operation)` 括号内部的三个表达式都可以省略 为什么在这里 `"for" "(" exprStmt expr? ";" expr? ")" stmt` 用到了两种不同的写法，如果是判断空语句的话 `exprStmt` 不是也可以做到吗

```
// 下面是 parse.stmt() 的处理内容

if (equal(Tok, "for")) {
// for ( i = 0; i < 10 ; i = i + 1) {...}

Node *Nd = newNode(ND_FOR);
// "("
Tok = skip(Tok->Next, "(");

// exprStmt
Nd->Init = exprStmt(&Tok, Tok);

// expr?
if (!equal(Tok, ";"))
  Nd->Cond = expr(&Tok, Tok);
// ";"
Tok = skip(Tok, ";");

// expr?
if (!equal(Tok, ")"))
  Nd->Inc = expr(&Tok, Tok);
// ")"
Tok = skip(Tok, ")");

// stmt
```

```
Nd->Then = stmt(Rest, Tok);
return Nd;
}
```