Os problemas que acarretam o uso de GOTO

Victor Barbosa Rocha

Nos primórdios da computação, quando não existiam funções como While, ifs e elses, os cientistas da época usavam estruturas chamadas GOTOs para fazer a lógica fazer a lógica dos seus programas. Contudo, essa ferramenta poderia causar problemas se mal utilizada e foi aos poucos sendo substituídas por estruturas mais elegantes para desempenhar saltos.

Um dos problemas desse tipo de programação é a complexidade para entender e até mesmo para provar que o que estava ali estava correto, ou seja, é problemático para debuggar. Esse tipo de código ficou conhecido como Spaghetti Code, fazendo alusão à forma como era estruturada os caminhos de dados num programa, cheio de linhas se cruzando. Dijkstra, famoso cientista de computação, advogou que “go to statements” são muito primitivos e que novas estruturas deveriam ser usadas para cumprir as mesmas tarefas dessa função, mas de forma sequencial, pois a possibilidade de saltar de um instrução para outra impossibilita a construção de um contexto sólido no programa e facilita a criação de um código não-limpo.

Outro problema acarretado pelo uso dessa função é a facilidade de ocorrências de loops infinitos que podem causar se o ponto de destino estiver acima da chamada da função. Por exemplo, se for utilizado um “catch-block” para pegar um erro e se esse “catch”não for cancelado e um erro ocorrer depois do ponto de retorno do GOTO, o programa entrará em loop.

Embora, GOTOs tenham um má reputação dentre as linguagens de programação, deixando até de existir em muitas delas, esse tipo de instrução ainda é muito utilizado em assembly, por exemplo o jump que funciona basicamente como um GOTO;