

Resumo sobre o artigo: “O Desenvolvimento da Linguagem C”

Rafael Rodrigues Oliveira

Belo Horizonte, 2023

Resumo:

O artigo foi escrito por Dennis Ritchie, o criador da linguagem C, com o propósito de abordar de forma detalhada o desenvolvimento dessa linguagem. Com essa finalidade, pretendo discutir a história, as origens e influências, a portabilidade, padronização e o sucesso da linguagem C.

1. História:

Nesta seção o autor fala sobre a história da linguagem C, descrevendo suas origens e as linguagens que a influenciaram.

No final dos anos 60, na Bell Labs, Ken Thompson queria criar um novo ambiente de computação, abandonando o problemático projeto Multics GE-645. Seu objetivo era desenvolver um sistema operacional com recursos inovadores, incluindo uma estrutura de arquivo em árvore, um interpretador de comandos, representação simples de arquivos de texto e acesso a dispositivos. Inicialmente, Thompson programou em montador para a máquina DEC PDP-7 de recursos limitados. O sistema evoluiu para o Unix, com uma linguagem de alto-nível chamada B, uma precursora da linguagem C. Mais tarde, a linguagem TMG de Doug McIlroy foi adaptada para o Unix. Esse desenvolvimento implicou na criação do Unix, uma das bases da computação moderna.

1.1 Origens e influências:

Como já sabemos, a linguagem C evoluiu de duas linguagens antecessoras, a BCPL e B. BCPL foi criada por Martin Richards na universidade de Cambridge, no Reino Unido e era destinada a criação de sistemas operacionais como o OS6 e o Xerox

PARC, mas a mesma era considerada muito limitada, lenta e complexa. E pensando nisso que Ken Thompson criou a linguagem B, uma simplificação de BCPL com novas funcionalidades.

BCPL, B e C se ajustam fortemente a família de linguagens imperativas, simbolizadas pelo FORTRAN e ALGOL60. Todas as três foram criadas para programação de sistemas operacionais, são pequenas e descritas compactamente, facilitando o trabalho de compiladores. E são consideradas linguagens “próximas da máquina”.

BCPL, B e C, possuem sintaxes distintas, mas de forma geral são muito parecidas. Um programa nessas linguagens consiste na criação de procedimentos e declarações globais. E a maior diferença de BCPL para B e C, está no fato de que em BCPL, pode-se declarar um procedimento dentro de outro, isso não é possível em C e B.

As mudanças chave na sintaxe de BCPL para B residem na notação de atribuição de valores, em que BCPL utiliza ':=' e B utiliza '='. Da mesma forma, as convenções de comentários também divergem: em B, utiliza-se '/* */' para comentários, enquanto BCPL faz uso de '/* */'. É importante observar que C segue a mesma convenção de B em relação à atribuição de valores e aos comentários.

O principal problema de BCPL e B, era que ambas eram endereçadas por palavra, então todo dado ocupava o mesmo espaço na memória, e isso acabava tornando o desenvolvimento nessas linguagens complicado.

1.2 Criação de C:

C foi desenvolvida como uma evolução necessária de B devido às limitações e problemas desta última. A motivação era criar uma nova versão do sistema operacional Unix, inicialmente denominada New B (NB). O desenvolvimento do NB teve início em 1971, sob a liderança de Dennis Ritchie. Ele começou a expansão de B incorporando o tipo "char". Essencialmente, o NB era considerado um estágio embrionário do C, representando uma extensão e aprimoramento do B original. Foi apenas em 1972 que Dennis Ritchie conseguiu finalizar o desenvolvimento da linguagem C.

2. Portabilidade e padronização:

Com a rápida popularização de C, após o lançamento do livro “The C Programming Language”, acabaram surgindo diversas variações de C, que embora fossem idênticas, eram constantemente incompatíveis entre si. A partir disso o comitê ANSI começou a padronização de C em 1983, o objetivo era criar uma linguagem que fosse independente de máquina, ou seja, que fosse portátil. Essa padronização foi finalizada em 1989 e ficou conhecida como ANSI C. Ela incluiu diversas

funcionalidades a linguagem, como bibliotecas padrão de: entrada e saída padrão, manipulação de strings, operações matemáticas complexas, etc.

Essa padronização acabou fazendo C se tornar uma linguagem extremamente portátil, que funciona na maioria de arquiteturas de computadores.

3. Sucesso:

Nesta seção, o autor enfatiza que C superou todas as expectativas iniciais em relação à linguagem. Ele destaca que C foi criada com um propósito extremamente específico em mente. O autor conclui o artigo argumentando que C, apesar de ser considerada uma linguagem artilosa e imperfeita, alcançou um sucesso notável devido à precisão com que cumpriu seu propósito fundamental que era criar a nova versão do UNIX.

Referência:

Ritchie, Dennis M. O desenvolvimento da linguagem C. Nova Jersey: Lucent Technologies Inc, 1996. Disponível em: <<https://www.bell-labs.com/usr/dmr/www/chistPT.html>>. Acesso em: 30/08/2023.

Apêndice:

C é uma linguagem de programação muito importante e influente na área da computação, sem ela, é bem possível que não teríamos muitas das tecnologias existentes atualmente, ou as mesmas seriam completamente diferentes das que conhecemos. Sua importância vem da capacidade de ser uma linguagem extremamente portátil, muito eficiente e de ter influenciado diversas linguagens de programação mais novas. A mesma foi criada no começo da década de 1970, mas é muito utilizada até os dias de hoje, por sua capacidade de adaptação a novas tecnologias, e isso é raro, porque LPs mais velhas vão ficando muito defasadas e acabam caindo no desuso, sendo substituídas por outras mais atuais.

Embora seja comumente escolhida para o ensino de programação devido à sua sintaxe simples e de fácil compreensão, a linguagem C é, na verdade, desafiadora de dominar, principalmente devido à necessidade de alocação manual de memória pelo programador. Essa tarefa pode acabar resultando em erros inesperados e difíceis de se depurar.