

Codificação em árvore binária

Renan Lemes Leepkaln¹

DEMAT/UNICENTRO, Guarapuava, PR

Jotair Kwiatkowski Jr.²

DECOMP/UNICENTRO, Guarapuava, PR

1 Introdução

Com o avanço tecnológico, a uma grande demanda tanto na área computacional quanto na matemática, com isso temos diversos problemas como a manipulação de dados, daremos ênfase a técnica usada como codificação binária de [1] que tem procedimentos simples, em que os dados do usuário vão estar organizados, e vão ser transformados e depois embaralhados para ter mais segurança no envio, nesse presente trabalho falaremos sobre tal assunto e discutindo sobre uma aplicação prática como ideia principal a nossa árvore binária, que é um tipo de estrutura que associa cada elemento formado vários subconjuntos seguindo de forma hierárquica, junto com nossa árvore usaremos a codificação da base binária de [3], que foi desenvolvida em (1930) por (Claude Sannon) que tem grande parte da representação de números naturais por meio de divisões sucessivas por dois. Analisando estudos sobre base binária podemos identificar imagens, palavras, números e diversos tipos de dados e isso tem ligação direta com a teoria matemática da informação, em que esse estudo procura contribuir na era digital.

2 Materiais e Métodos

A implementação do código se baseia em dados homogêneos, ou seja, o usuário digita qualquer tipo de entrada para o código podem ser números, letras ou caracteres especiais. Sabemos que por conta disso o desenvolvimento então temos muitos sistemas amarrados ao nosso cotidiano que está exatamente ligado a e-mails, contas bancárias, etc... ou seja, a tecnologia ficando mais vinculada no nosso dia a dia, desta forma a codificação auxilia na segurança dos dados. Neste trabalho usaremos a base binária ela mostra que é possível escrever qualquer tipo de caracteres do teclado com números com apenas um e zero, assim partiremos para o código que simula uma entrada do usuário como uma senha ou palavra-chave, que é alocada em um vetor, e depois da entrada dividiremos esse vetor em partes, para separá-las, e assim transformaremos em números da tabela (ASCII). Onde cada parte do vetor de entrada se torna um número tabelado com a linguagem de máquina, depois alocaremos em nossa árvore formando uma hierarquia dependendo da raiz sendo a primeiro número da nossa entrada digitado pelo usuário, sendo nós menores que o valor alocado na raiz serão alocados para esquerda e maiores alocados para a direita dessa forma temos uma árvore binária como mostra a figura [1].

¹renanl@hotmail.com

²jotairkw@yahoo.com.br

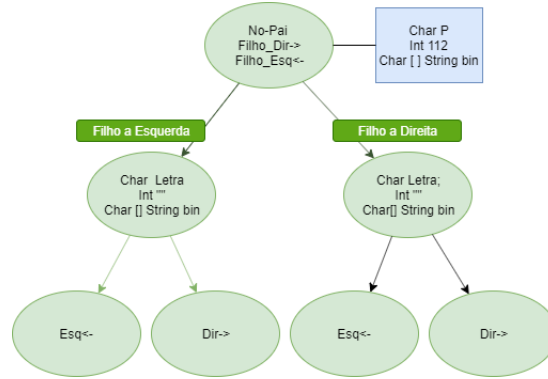


Figura 1: Separação das entrada

3 Resultados e Discussões

Considerando nossa simulação de pequeno porte, obtivemos sucesso e efetividade do código sem problemas ou erros, porém com a diversidade de caracteres do teclado pode ocorrer erros de repetição caso usuário tenha repetido caracteres, porem para corrigir esse erro generalizamos a árvore, para caracteres repetidos assim tendo uma sequência hierárquica como [2] nova e correta, que funciona como maior e igual para direita e menores para esquerda. E como podemos ver na tabela [1] percebemos uma grande diferença probabilística, que é devido as permutações pois quando separamos os caracteres temos uma função fatorial que no caso nossa senha se torna um anagrama, e transformando-os em binários impossibilitaria ainda mais que obtenham a senha do usuário. Em trabalhos futuros trabalharemos com a hierarquia da árvore assim problematizando ainda mais nossos dados, e tendo melhor eficácia na segurança.

Tabela 1: Coeficientes de possibilidades de eventos.

| Caracteres | Arranjo normal | Arranjo da árvore | Árvore e base binária |
|------------|----------------|-------------------|-----------------------|
| 1 | 0,78125 | 0,78125 | 0,78125 |
| 5 | 3,968753981 | 3174796800 | 125999874400 |
| 10 | 8,101549991 | 8,23179E+20 | 6,66903E+21 |
| 12 | 8,949007618 | 1,13648E+25 | 1,01704E+26 |
| 14 | 10,9375 | 1,51607E+29 | 1,6582E+30 |
| 16 | 11,53534266 | 1,953E+33 | 2,25285E+34 |

Analisando a tabela a cada caracter que aumenta ela vai se torna mais difícil de descobrir a entrada do usuário, no caso da árvore ai ela se torna um anagrama fica fatorialmente maior.

Referências

- [1] S. Robert. *C Completo e Total*. 3 Edição. São Paulo: Makron Books ltda, 1997.
- [2] B. Francisco. *Estrutura de dados e técnicas de programação*. 1 Edição. Rio de Janeiro:Elsevier Editora Ltda, 2014.
- [3] M.L. Herman. *Números binários em livros didáticos de matemática e de computação: uma comparação*. Ponta Grossa: RBECT, 2018.