

Projeto – Mini Googlebot (Parte I)

Data de entrega: 15/10. Será descontado 1 ponto por dia de atraso.

Grupos de 2 alunos

1. Objetivo

O objetivo deste projeto é desenvolver um Mini Googlebot, isto é, uma simulação do algoritmo Google real para captação de informações. O Google utiliza um robô, chamado Googlebot, para varrer a web em busca de links que apontem para sites. O resultado é um conjunto de informações de cada site, permitindo que o algoritmo de busca do Google faça computações sobre as mesmas decidindo a relevância (ou não) de cada site para a busca do usuário. O projeto deverá então utilizar estruturas de dados adequadas para armazenar (em memória) e processar as informações de modo eficiente.

2. Especificação

O sistema Mini Googlebot considera que cada site possui as seguintes informações:

- Código: valor numérico com 4 dígitos (chave primária);
- Nome: string de até 50 caracteres;
- Relevância: valor numérico entre 0 e 1000. Zero indica a menor relevância e mil a relevância máxima;
- Link da página principal: string de até 100 caracteres;
- Palavras-chave: cada site pode ter no máximo 10 palavras-chave, sendo cada uma delas uma string de até 50 caracteres.

Tais informações devem ser lidas de um arquivo texto, chamado “googlebot.txt”, no formato CSV (*Comma Separated Value*) – os dados do site em cada linha são separados por vírgulas. Cada grupo deve criar um arquivo googlebot.txt, com entradas suficientes para testar adequadamente o

programa desenvolvido. Fica implícita a necessidade de desenvolver funções para leitura do arquivo e para alimentar as estruturas de dados. Abaixo um exemplo de arquivo com 2 sites:

0001,YouTube,900,https://www.youtube.com/,youtube,videos,online 0002,Netflix,800, https://www.netflix.com/br/,netflix,videos,streaming

A seguir, apresentam-se as operações do sistema Mini Google que devem ser implementadas nesta primeira etapa do projeto. Cabe ao grupo definir as melhores estruturas de dados a serem utilizadas.

Ao ser executado, o programa deve apresentar um menu com as seguintes opções:

1. **Inserir um site:** a inserção deve ser ordenada pela chave primária, isto é, o código do site. O sistema não deve aceitar valores de código já existentes;
2. **Remover um site:** dado um código, deve-se excluir o site. Caso um código inexistente seja inserido, uma mensagem deve ser printada ao usuário a respeito disso;
3. **Inserir palavra-chave:** dado um código, pode-se adicionar uma nova palavra-chave.
4. **Atualizar relevância:** dado um código, deve-se atualizar a relevância do site;
5. **Sair:** finalizar o programa.

3. Ferramentas, modelo de desenvolvimento e de entrega

O projeto deve ser desenvolvido em grupos de 2 alunos. A implementação deve ser realizada em linguagem C, padrão C99, ou seja, com a flag `-std=c99` passado ao compilador.

Deve ser submetido um arquivo compactado (.zip) no RunCodes de um dos alunos do grupo. O arquivo .zip deve conter: os fontes (arquivos .c e .h), um Makefile que compile o projeto e um arquivo README. O projeto pode estar organizados em pastas/diretórios, desde que o Makefile seja configurado para fazer a compilação adequadamente. O arquivo README deve conter: os nomes de todos os integrantes do grupo; compilador/plataforma utilizado; instruções de uso do programa e justificativas para as escolhas das estruturas assim como justificativas para as implementações escolhidas para as estruturas e de seus métodos (inserção, remoção e busca).

4. Critérios de Avaliação

- Aderência à especificação e ao uso de TADs (5 pontos)
- Eficácia e eficiência da solução (4 pontos)
- Documentação (1 ponto)

- Funcionalidades (facilidades de uso do programa – 1 ponto) (Bônus – Importante: o bônus só é válido para os grupos com nota máxima em aderência à especificação)

Observações:

- 1- Os projetos serão compilados e corrigidos utilizando o compilador GCC no Linux.
- 2- É óbvio ululante que haverá verificação de plágio. Aos projetos envolvidos em plágio serão atribuídos nota zero.

Bom Trabalho!