

INSTITUTO DE ENGENHARIA E TECNOLOGIA - IET

Curso: Ciência da Computação

Disciplina: Programação e Estruturas de Dados

Professora: Jaqueline Faria de Oliveira

Valor: 10 pontos

TRABALHO PRÁTICO 1

1. Cache web – Histórico de Acessos na Internet

Em redes de computadores, um proxy (em português procurador) é um servidor (um sistema de computador ou uma aplicação) que age como um intermediário para requisições de clientes solicitando recursos de outros servidores. Um cliente conecta-se ao servidor proxy, solicitando algum serviço, como um arquivo, conexão, página web ou outros recursos disponíveis de um servidor diferente e o proxy avalia a solicitação como um meio de simplificar e controlar sua complexidade. Os proxies foram inventados para adicionar estrutura e encapsulamento a sistemas distribuídos. Hoje, a maioria dos proxies são proxies web, facilitando o acesso ao conteúdo na *World Wide Web* e fornecendo anonimato.

Uma aplicação proxy popular é o proxy de armazenamento local (ou *cache*) web, em inglês *caching web proxy*, um proxy web usado para armazenar e atualizar (conforme pré-programado). Este provê um armazenamento local de páginas da Internet e arquivos disponíveis em servidores remotos da Internet assim como sua constante atualização, permitindo aos clientes de uma rede local (LAN) acessá-los mais rapidamente e de forma viável sem a necessidade de acesso externo.

Quando este recebe uma requisição para acesso a um recurso da Internet (a ser especificado por uma URL), um proxy que usa cache procura resultados da URL em primeira instância no armazenamento local. Se o recurso for encontrado, este é consentido imediatamente. Senão, carrega o recurso do servidor remoto, retornando-o ao solicitante que armazena uma cópia deste na sua unidade de armazenamento local. O cache usa normalmente um algoritmo de expiração para a remoção de documentos e arquivos de acordo com a sua idade, tamanho e histórico de acesso (previamente programado). Dois algoritmos simples são o *Least Recently Used* (LRU) e o *Least Frequently Used* (LFU). O LRU remove os documentos que passaram mais tempo sem serem usados, enquanto o LFU remove documentos menos frequentemente usados.

2. Descrição do trabalho prático

Deve ser criado um programa que simule o funcionamento de um histórico de Páginas de um Cache. Considere que o Cache tem tamanho máximo de 10 elementos no seu Histórico de Acessos. Quando esse histórico está cheio uma das páginas anteriores deve ser excluída. Devem ser implementados dois Históricos, um deve obedecer às regras de exclusão LRU e outro com regras LFU. Deseja-se ainda armazenar um log que contenha todas as 20 páginas acessadas pelo usuário, caso o log esteja cheio, devem ser apagados os acessos mais antigos. O usuário deve inserir manualmente as páginas para essa simulação. Podem ser inseridas quantas páginas forem necessárias, podendo estas ser repetidas ou não. A cada vez que uma página for informada o programa deve enviá-las para os dois Históricos e também para o LOG e depois exibi-los.

Ao enviar para cada histórico deve-se verificar:

- Histórico com regras LRU:
 - Se página já está no histórico, atualizar data de acesso.
 - Ao atualizar a página esta deve se tornar a primeira do histórico.
 - Se página não está no histórico, inserir essa página.
 - Caso o histórico esteja cheio, deve-se excluir um item do histórico conforme regras do algoritmo LRU.
- Histórico com regras LFU:
 - Se página já está no histórico, atualizar a quantidade de acessos.
 - Ao atualizar a página esta deve se tornar a primeira do histórico.

INSTITUTO DE ENGENHARIA E TECNOLOGIA - IET

Curso: Ciência da Computação

Disciplina: Programação e Estruturas de Dados

Professora: Jaqueline Faria de Oliveira

Valor: 10 pontos

- Se página não está no histórico, inserir essa página.
- Caso o histórico esteja cheio deve-se excluir um item do histórico conforme regras do algoritmo LFU.
- LOG
 - A página deve ser inserida ao final do log, mesmo que esta já tenha sido acessada.
 - Caso o log esteja cheio, deve-se retirar o log mais antigo.

Devem ser dadas ao usuário as seguintes opções:

1. Inserir página no histórico
2. Listar Histórico LRU
3. Listar Histórico LFU
4. Lista LOG
5. Limpar históricos e LOG
6. Sair

Implementação:

Crie um algoritmo que implemente uma classe Página:

Classe Pagina:

- ID - Integer
- Título da página – String
- URL – String
- Data Acesso – Datetime.
- Quantidade de Acessos – Integer

Deve ser implementado o nodo do tipo Página. Esse nodo deve ser utilizado nos TAD Pilha e Fila que serão utilizados na implementação.

Deve-se utilizar as TADs Pilha e Fila implementadas com Lista Encadeadas, conforme visto em sala de aula. Devem ser utilizadas duas pilhas para implementação, uma dos históricos LRU e para os históricos LFU, e Fila para o LOG.

Regras para implementação das pilhas:

- A pilha deve se manter sempre organizada por data de acesso. Desta forma, quando uma página já faz parte do Histórico e é acessada novamente deve-se colocar a página no topo da pilha e atualizar sua data de acesso. Isso deve funcionar para ambas as pilhas.
- A fila deve se manter organizada por data de acesso do histórico, ou seja, no início terão os logs das páginas acessadas a mais tempo e no final da fila as mais recentes. Deve-se inserir o log sempre no final da Fila e retirar sempre no início.
- Demais regras de implementação devem obedecer às regras do Tipo Abstrato de Dados Pilha e Fila passados em sala (LIFO).

Devem ser feitas as seguintes validações:

- Não deve ser possível inserir dois históricos com mesmo ID.

Instruções:

- Trabalho pode ser feito no máximo em dupla;
- Data de entrega: 25/10/2017 até 23h59;
- Deve ser enviado somente um arquivo com os arquivos do projeto compactados;
- O arquivo deve ser enviado para a atividade Trabalho Prático 1 na Sala de Aula Virtual;
- Serão avaliados:
 - Indentação;

INSTITUTO DE ENGENHARIA E TECNOLOGIA - IET

Curso: Ciência da Computação

Disciplina: Programação e Estruturas de Dados

Professora: Jaqueline Faria de Oliveira

Valor: 10 pontos

- Clareza do código;
- Utilização das estruturas solicitadas;
- Resolução correta do problema proposto;
- Documentação do algoritmo através de comentários.
- **Atenção: Qualquer identificação de cópia e/ou plágio os trabalhos que estiverem nessa situação receberão nota 0 (zero)**

Bom trabalho a todos! ☺

Professora Jaqueline