

ATIVIDADE PRÁTICA – 1 - CACHES - 2023 - 1

Esta atividade avaliativa pode ser realizado em grupos de até 4 alunos. **Grupos com mais de 4 alunos irão provocar a anulação da atividade. Esta atividade tem peso no cálculo da média conforme explicitado no Plano de Ensino.** Você deve ler todo documento antes de começar e considerar o seguinte código de ética: *você poderá discutir todas as questões com seus colegas de classe, professores e amigos. Poderá também consultar os livros de referência da disciplina, livros na biblioteca virtual ou não, e a internet de forma geral e abrangente nos idiomas que desejar. Contudo, o trabalho é seu e deverá ser realizado por você. Cópias ensejarão a anulação do trabalho.*

OBJETIVO

Pesquisar e praticar. Pesquisar os conteúdos que irão complementar o material apresentado em sala, ou nos livros sugeridos na ementa, e praticar estes mesmos conceitos. Esta é uma oportunidade para aprimorar sua formação e se destacar no mercado. Uma avaliação com oportunidade de crescimento acadêmico e profissional.

DESCRIÇÃO DO TRABALHO

A Empresa **Texto é Vida**, pediu que você desenvolvesse, usando a linguagem de programação Python, em um ambiente virtual como o Repl.it ou o Google Colaboratory um aplicativo de leitura dos seus 100 textos mais importantes que fosse flexível, rápido e eficiente. Infelizmente, por motivos legais, estes textos só podem ser fisicamente armazenados em um sistema de discos forense que é muito lento e como o menor destes texto tem mais de 1000 palavras, muito tempo é gasto para durante o processo de abertura, carregamento do arquivo e apresentação em tela para leitura.

O seu objetivo será criar um aplicativo de leitura de textos, no terminal, que minimize o tempo de carregamento dos textos físicos em memória e indicar qual o algoritmo de cache mais eficiente para atender as demandas da empresa. Para isso você terá que seguir as seguintes instruções:

1. Precisar ter exatamente 100 textos diferentes armazenados em um disco, mesmo que este disco esteja em um ambiente virtual, identificando cada texto com um número inteiro.
2. Precisar criar uma estrutura de memória capaz de armazenar, no máximo dez destes textos.
3. Precisar testar, no mínimo três algoritmos de cache diferentes.

A interface entre o usuário e o seu sistema será muito simples. Ao rodar, o programa pergunta o número que identifica o texto desejado, abre este texto para leitura e fica aguardando o número de um novo texto. Se o número for zero o programa deverá ser encerrado. Se o número for -1 o programa deverá entrar em modo de simulação.

Todos os arquivos serão carregados no cache antes de ser apresentado ao usuário, a não ser que este arquivo já esteja no cache. Lembre-se o tempo de apresentação do arquivo de textos em tela deve ser considerado como fundamental pelo seu programa.

Modo de Simulação

O modo de simulação será ativado pela entrada -1 na linha de comando.

Para escolher os algoritmos de cache, seu gestor de projeto solicitou que você simulasse o acesso de 3 usuários aleatórios por algoritmo de cache e medisse o tempo gasto entre a solicitação do arquivo e a apresentação deste arquivo no terminal (*cache hit e time*), e o número de vezes que um determinado arquivo foi solicitado e não estava no cache (*cache miss e time*). O gestor de projeto também solicitou que cada usuário aleatório fizesse, no mínimo 200 solicitações de arquivos e em cada uma, o número do arquivo desejado fosse escolhido de três formas diferentes: aleatório, puro e simples, aleatório com distribuição de poison e aleatório de forma que textos numerados entre 30 e 40 tenham 33% de chance de serem sorteados.

A saída desejada, relatório do modo de simulação, é um arquivo de texto formatado de forma a apresentar o tempo médio gasto para apresentar cada texto e o número de vezes em que o arquivo solicitado não estava no cache considerando os usuários aleatórios e os modos de sorteio dos arquivos. Este relatório deve ser criado de forma que seja possível identificar cada um dos usuários, cada um dos algoritmos de cache, os tempos de apresentação de cada texto e o número de vezes que o arquivo solicitado não estava no cache e o número de vezes em que o arquivo estava no cache.

Para criar os 100 arquivos de textos diferentes, com mais de 1000 palavras, você pode, por exemplo, baixar livros disponibilizados pelo Projeto Gutenberg em ([Browse By Language: Portuguese | Project Gutenberg](#)) e dividir estes o arquivo em textos menores de 1000 palavras. **Lembre-se o uso de qualquer arquivo de texto, ou site, que viole os direitos autorais irá provocar o zeramento da atividade.**

Avaliação

Este trabalho será avaliado da seguinte forma:

1. Todas as funcionalidades solicitadas rodam, foi possível encontrar todos os textos no ambiente *online*, os textos têm no mínimo 1000 palavras e o relatório de simulação apresentado atende o solicitado. Nota 10.
2. Todas as funcionalidades solicitadas rodam, Contudo, não foram encontrados todos os textos, ou os textos não têm no mínimo 1000 palavras, ou existe algum problema com o relatório: Nota 8.

3. Algumas funcionalidades rodam e o relatório dos algoritmos de cache está correto: Nota 7.
4. Algumas funcionalidades rodam. Mas, existem problemas no relatório dos algoritmos de cache: Nota 5.
5. Qualquer outra situação diferente das descritas acima: Nota 0.