



Prof: Nilton Luiz Queiroz Junior  
Disciplina: Programação Concorrente (5205 – Turma 1)

## Primeiro Trabalho Prático

**Objetivo:** O trabalho tem como objetivo a implementação de uma aplicação paralela, em um ambiente de memória compartilhada centralizada e a elaboração de um relatório, fazendo análises de desempenho da aplicação implementada.

### Instruções:

- O trabalho poderá ser feito em grupos de 1 a 3 membros;
- A implementação da aplicação deverá ser feita na linguagem C/C++ utilizando a biblioteca *pthread*;
- O trabalho tem valor de 0 a 10 e corresponde à primeira avaliação periódica da disciplina;
- Deverão ser entregues os seguintes itens:
  1. O código fonte do trabalho;
  2. Um passo a passo de como compilar e executar o programa;
    - Como deve ser fornecida a entrada para o programa, como escolher a quantidade de *threads* a serem usadas, qual sistema operacional a ser usado, e o que precisa ser instalado, como instalar e como configurar, etc;
  3. Um relatório, de 4 a 6 páginas, fazendo análise de desempenho composto por:
    - Introdução;
    - Referencial teórico;
      - Descrever superficialmente técnicas que foram usadas na paralelização do algoritmo:
        - Por exemplo:
          - Descrever os conceitos de threads mecanismos de sincronização utilizados;
          - Etc;
    - Desenvolvimento;
      - Explicar como funciona a aplicação e como ela foi paralelizada;
    - Ambiente experimental e experimentos realizados;
      - Descrever hardware e sistema operacional utilizado;
      - Descrever detalhes dos experimentos que foram realizados;
        - Tamanho das entradas, quantidade de threads, parâmetros do algoritmo;
    - Análise e discussão dos resultados;
      - Extrair métricas tais como speedup, entre outras vistas em sala;
      - Se desejável colocar gráficos (ou tabelas) com speedup por quantidade de threads;
    - Conclusões;
  - Os itens acima deverão ser entregues via moodle, em um arquivo compactado que deve seguir o padrão:
    - NomeAluno1\_RAXXXXXX\_NomeAluno2\_RAXXXXXX\_NomeAluno3\_RAXXXXXX.zip;
      - O formato do arquivo para submissão poderá ser .ZIP, .RAR ou .TAR
  - A data limite para a entrega do trabalho será combinada em sala de aula;



Prof: Nilton Luiz Queiroz Junior  
Disciplina: Programação Concorrente (5205 – Turma 1)

**Descrição:** Neste trabalho deverá ser implementada uma versão simplificada do algoritmo K-NN, fazendo a paralelização dos cálculos realizados sobre os dados da base de conhecimento, onde a classe será dada somente pelo vizinho mais próximo, ou seja, deverá ser implementado o algoritmo K-NN para  $K=1$ . Além disso, não é necessário fazer a retenção do novo conhecimento obtido.

Serão disponibilizadas 5 bases de conhecimento, todas elas contendo exemplos no contexto de reconhecimento de gêneros musicais. Cada base tem uma quantidade diferente de atributos, porém todas elas seguem o mesmo padrão:

- Cada linha representa um exemplo;
- Todos os atributos, exceto a classe, de cada exemplo são números reais;
- Os atributos são todos separados por vírgula;
- O último atributo é a classe;
- A classe é uma string;

Para os experimentos, além das bases de conhecimento, serão também disponibilizados alguns exemplos de teste para cada base. Os arquivos de teste seguem o mesmo padrão da base, incluindo a classe de cada exemplo, porém as classes para os exemplos de teste devem ser ignoradas durante a predição da classe. Se o grupo desejar poderá avaliar a precisão do método na tarefa de classificação, porém este não é o foco do trabalho.

As bases de conhecimento tem a seguinte nomenclatura:

train\_<numero\_de\_atributos>.data

Os arquivos com os exemplos de treino tem a seguinte nomenclatura:

test\_<numero\_de\_atributos>.data

Por exemplo:

train\_256.data

test\_256.data

Esses dois arquivos indicam que a base tem 256 atributos excluindo a classe, ou seja, são 256 números reais, e após tais números se encontra a string que indica a classe.

Para um melhor entendimento do algoritmo K-NN, a seguir serão apresentados alguns conceitos de aprendizagem de máquina e uma pequena descrição do algoritmo.

Dentro da área de aprendizagem de máquina existem algoritmos que apenas retém em uma base de conhecimento alguns exemplos, com suas classes já conhecidas, para predizer a classe de novos exemplos, tais algoritmos são também chamados de algoritmos de aprendizagem baseada em instâncias. Um desses algoritmos é o *k-nearest neighbors* (K-NN), que procura os  $k$  exemplos mais próximos do exemplo que se deseja classificar.

Cada exemplo da base de conhecimento é formado por um conjunto de atributos, dessa forma, um exemplo pode ser visto como um vetor. Dentro desses atributos existe um deles que é especial, tal atributo é denominado classe, que descreve o fenômeno que se deseja aprender.

Assim, para se classificar um novo exemplo X usando o algoritmo K-NN é necessária uma base de conhecimento B e também os atributos do novo exemplo.

Um pseudo-código para o algoritmo de classificação do K-NN pode ser dado a seguir:



Prof: Nilton Luiz Queiroz Junior  
Disciplina: Programação Concorrente (5205 – Turma 1)

Classificar(B, X, k) Para cada exemplo $E \in B$ Calcular a distância entre E e X Obter as classes das k menores distâncias Retornar a classe que mais apareceu dentre as k classes obtidas
<i>Pseudo-código para classificação de um exemplo no algoritmo K-NN</i>

Onde B é a base de conhecimento contendo os exemplos já classificados e X é o exemplo o qual será classificado e k é a quantidade de exemplos que influenciarão na predição da classe do exemplo X. É importante ressaltar que no cálculo da distância não pode ser usado o atributo de classe. Além disso, o cálculo de distância pode ser feito de diversas maneiras, sendo as mais comuns a distância Euclidiana e a distância Manhattan.

Distância Euclidiana entre dois pontos:

$$D(P, Q) = \sqrt{\sum_{i=1}^n (P_i - Q_i)^2}$$

Distância Manhattan entre dois pontos:

$$D(P, Q) = \sum_{i=1}^n |P_i - Q_i|$$

## Problemas com Trabalhos COPIADOS:

Quem copiar terá o trabalho anulado (zerado), seja de outra dupla ou da internet.  
Quem fornecer a cópia também terá o trabalho anulado (zerado).

### Referências para algoritmos de aprendizagem baseada em instância:

Stuart Russell and Peter Norvig. 2009. Artificial Intelligence: A Modern Approach (3rd ed.). Prentice Hall Press, Upper Saddle River, NJ, USA.  
Capítulo 18 seção 8 (Nonparametric models).

Ian H. Witten and Eibe Frank. 2005. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.  
Capítulo 4 seção 7