

# Sistemas Operacionais

## EP2 - outubro/2016

...

Carlos Augusto Motta de Lima - 7991228

Danilo Aleixo Gomes de Souza - 7972370

# Vetor Pista

O vetor pista é representado com uma variável global e como todo o sistema tem acesso a ele, um semáforo foi criado para lidar com as condições de corrida.

Além disso, uma barreira foi criada durante a execução das *threads* ciclistas, para garantir que primeiro todas as *threads* retirem a posição onde estão para somente depois marcarem sua nova posição no vetor pista e assim garantir que valores não sejam sobrescritos.

O vetor suporta apenas dois ciclistas na mesma posição sendo que uma das posições marca realmente onde o ciclista está e a segunda serve apenas para ultrapassagem.

# Threads Ciclistas

Cada ciclista é uma thread individual que é responsável por atualizar sua posição no vetor pista.

Quando algum ciclista termina a prova, ele é adicionado uma pilha que contém informações, referentes a sua colocação e seu tempo de chegada, por mais que a corrida acabe, ou seja, o terceiro ciclista de alguma equipe cruze a linha de chegada, o programa ainda sim espera que os outros ciclistas terminem a prova, para que em seguida seja impresso a colocação geral.

# Barreiras

O programa usou barreiras para garantir sincronia das threads, portanto existem três barreiras essenciais para o programa, são elas:

- Barreira\_todos\_ciclistas\_correram: a barreira é usada para impedir que o coordenador comece a execução antes de todos os ciclistas executarem e atualizarem sua posição.
- Barreira\_apagar\_posicao: a barreira é usada para impedir que alguma thread ciclista sobrescreva a posição de outra, fazendo com que primeiro todas as posições sejam apagadas do vetor pista, para que em seguida, as novas posições sejam colocadas.
- Barreira\_coordenador: a barreira permite que o coordenador execute as funções verificadoras, sem as threads ciclistas estarem rodando ao mesmo tempo e modificando as variáveis

# Experimentos

# Experimentos

Para a realização dos experimentos, foi utilizada uma máquina com dois processadores físicos e dois processadores virtuais (*hyperthreading*).

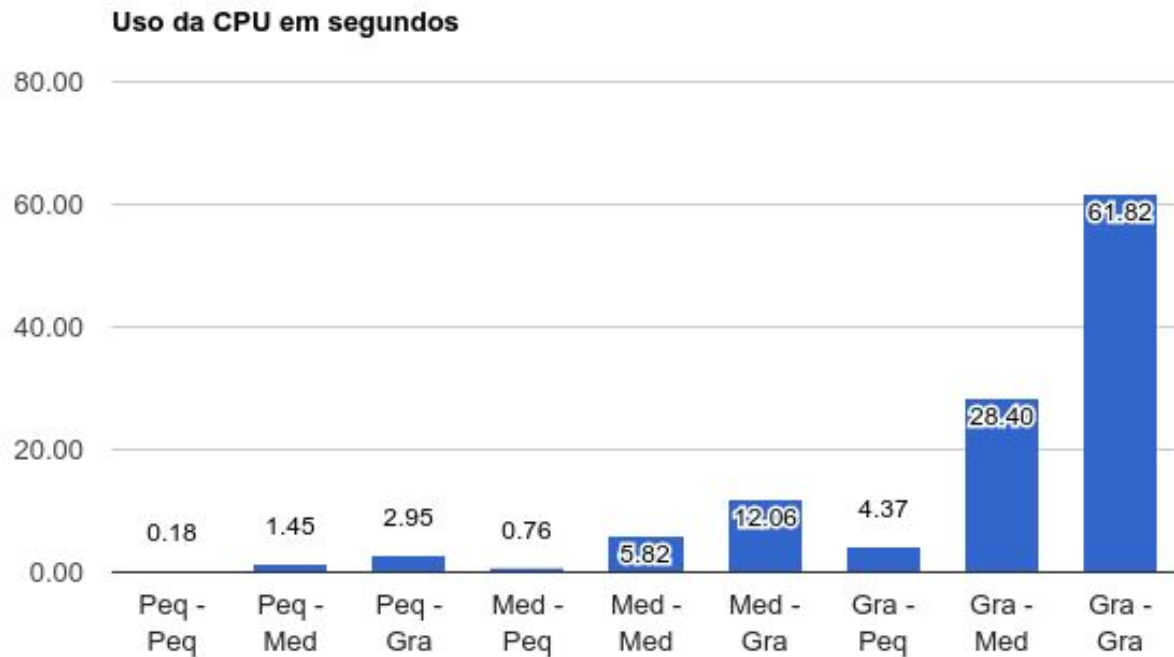
O tamanho da pista foi dividido em 250 metros para o tamanho pequeno, 1000 para o médio e 5000 para o grande.

Já os ciclistas foram divididos em poucos: 5, médios: 30 e muitos: 60.

Os resultados foram obtidos da função `/usr/bin/time -v` do Linux.

\* todos os resultados possuem dois parâmetros na legenda, o primeiro sempre se refere ao tamanho da pista e o segundo ao número de ciclistas, sendo peq. = pequeno, med. = médio e gra. = grande.

# Uso da CPU



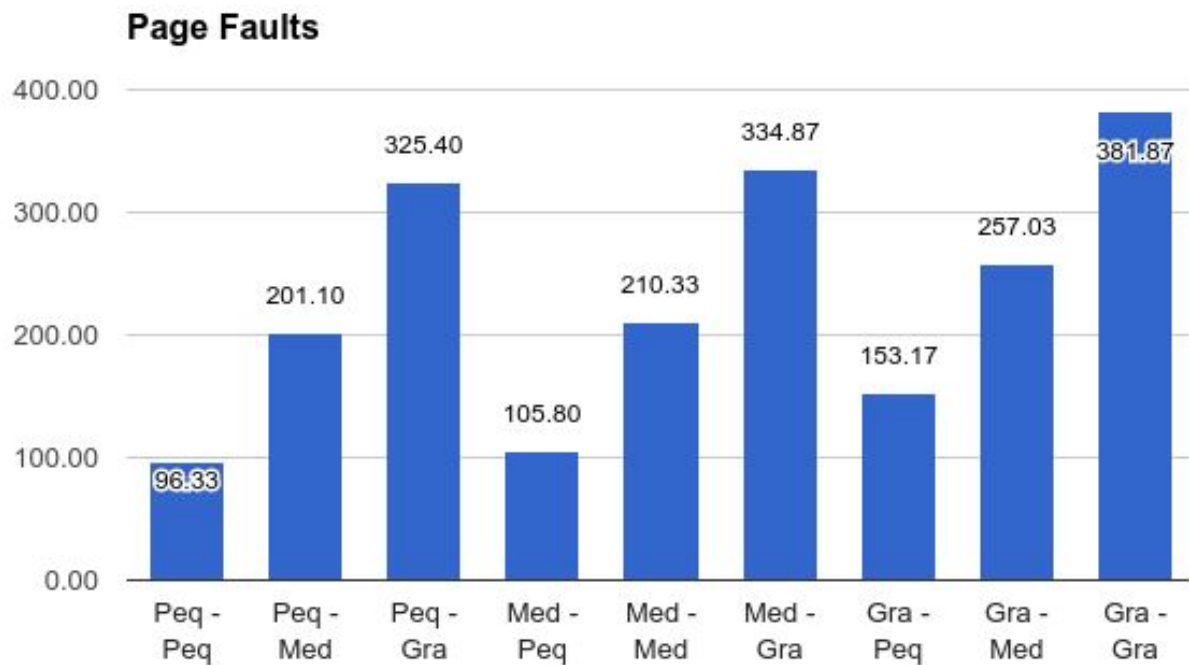
# Uso da CPU

Podemos observar que como esperado, quanto maior o tamanho da pista e maior a quantidade de *threads*, maior é o tempo de processamento do programa.

Quanto maior o tamanho da pista, mais interações o programa terá que fazer para cada ciclista dar uma volta na pista, também que quanto maior o número de ciclistas, mais processamento será necessário para cada passo ser dado na pista, pois com  $n$  ciclistas teremos que fazer  $n$  execuções de função que avança uma posição na pista.



# Page Faults

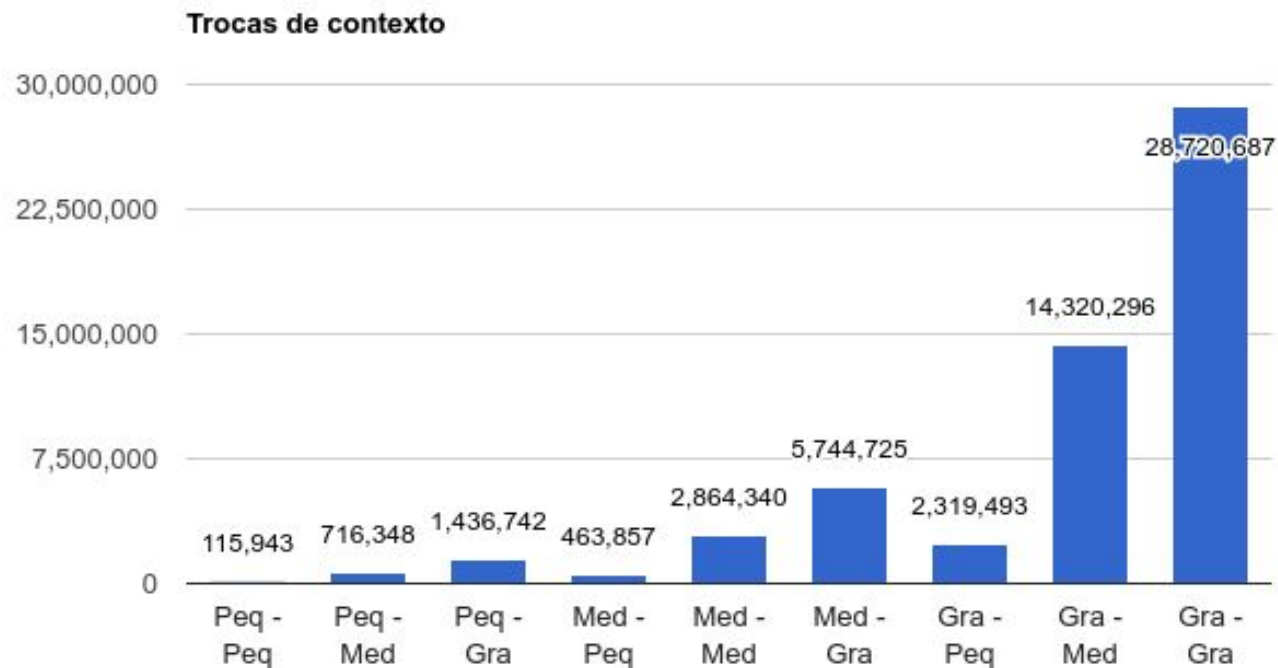


# *Page Faults*

No caso da contagem de *page faults*, podemos observar que ela aumenta de acordo com o número de ciclistas e se mantém, razoavelmente, de acordo com o tamanho da pista.

A explicação para este dá-se pelo fato de que quanto mais *threads*, mais trocas de contexto serão realizadas e portanto é mais provável que o espaço de memória que ela precise não esteja na memória física, visto que várias *threads* já rodaram desde a última vez que essa *thread* tinha rodado.

# Trocas de contexto



# Trocas de contexto

Podemos analisar o gráfico e analisar que as trocas de contexto no nosso programa aumentam, tanto em relação ao tamanho da pista como em relação ao número de ciclistas pelo fato de que as barreiras na biblioteca *pthread*, ao serem ativadas, cedem o processamento à outra *thread*, causando uma troca de contexto e por isso o aumento em relação às duas variáveis.