

EP2 - Corrida *Miss and Out*

Sistemas Operacionais (MAC0422)

Nome: Lara Ayumi Nagamatsu NUSP: 9910568

Nome: Lucy Anne de Omena Evangelista NUSP:11221776

Visão Geral

Este EP trata da simulação de uma corrida *Miss and Out* de threads ciclistas - ou seja, paralelizadas, mas de forma que possam escrever nas posições da pista sem a possibilidade de acessar um mesmo espaço de memória ao mesmo tempo.

Para controlar esse acesso, incluímos um mutex na função de criação de thread. Isso elimina a possibilidade de se sobrescrever posições da pista ou perder informações.

Ainda, visando o controle de cada instante/rodada, utilizamos a `main()` com a thread principal como escalonadora para o controle do tempo. Dessa forma, utilizamos uma variável (`ciclistas_atuais`) que seria incrementada a cada thread atualizada. Um sinal (`pthread_cond_t`) foi utilizado para regular quando as threads poderiam ser liberadas novamente.

Decisões de Projeto

- **QUEBRA DE CICLISTAS**

Os ciclistas recebem a possibilidade de quebrar a cada volta múltipla de 6, caso hajam 5 ciclistas correndo. Decidimos que ocorre a verificação de quantos ciclistas correm na dada rodada, e não quantos ciclistas ainda estão ativos no programa.

É importante notar que a determinação da quebra dos ciclistas ocorre na volta anterior à múltipla de 6, no momento em que o ciclista irá mudar de volta. Fazemos isso para não ocorrerem condições de corrida ou “filtragem” errada de quantos ciclistas devem passar para a próxima rodada.

Decisões de Projeto

- **ALEATORIEDADE NA ELIMINAÇÃO**

O projeto exige que o ciclista a ser eliminado nas rodadas pares seja escolhido aleatoriamente entre outros que chegarem ao mesmo tempo. Determinamos que a escolha seria feita pelo escalonador, ou seja, qualquer que fosse o ciclista que atualizasse sua posição por último (dentro do mesmo intervalo de tempo) seria o deletado;

- **ULTRAPASSAGEM**

A ultrapassagem pode ser feita se houver alguma brecha de caminho na fileira do ciclista, ou seja, uma posição livre do lado e a frente desta, independente ou não de haverem concorrentes dos lados. Assim que o ciclista consegue passar para a próxima faixa, pela esquerda ou direita, ele busca ficar o mais próximo da sua posição anterior.

Decisões de Projeto

- **DUAS ÚLTIMAS VOLTAS**

Para determinar a possibilidade de alteração velocidade para 90km/h nas últimas duas voltas, criamos um vetor de 2 posições que contém os finalistas. Esse vetor é preenchido no momento em que ocorre a “filtragem” para as últimas 2 voltas.

É importante notar que quando ocorre a mudança de um dos ciclistas conter a velocidade de 90km/h para nenhum dos dois últimos finalistas estar nessa velocidade, as threads de voltas anteriores que estavam, antes, bloqueadas por várias barreiras (a fim do timing de 20ms estar correto), são liberadas. Na próxima rodada, os ciclistas correm como esperado, se nenhum dos finalistas tornar a 90km/h novamente.

Decisões de Projeto

- **RANKEAMENTO**

Para registrar as **posições dos ciclistas por rodada** utilizamos um sistema de entrada de ciclistas, onde assim que algum ciclista inicia uma nova rodada ele é inserido no ranking da rodada. O ranking da rodada fecha quando a quantidade ideal de ciclistas acrescido de 1 passou pela rodada, com os devidos descontos de ciclistas quebrados. O último ciclista a entrar no ranking em uma rodada par é eliminado.

Para registrar as **posições dos ciclistas na corrida completa** utilizamos um sistema de saída, onde cada ciclista eliminado é registrado no ranking, com seu devido tempo de saída ou rodada.

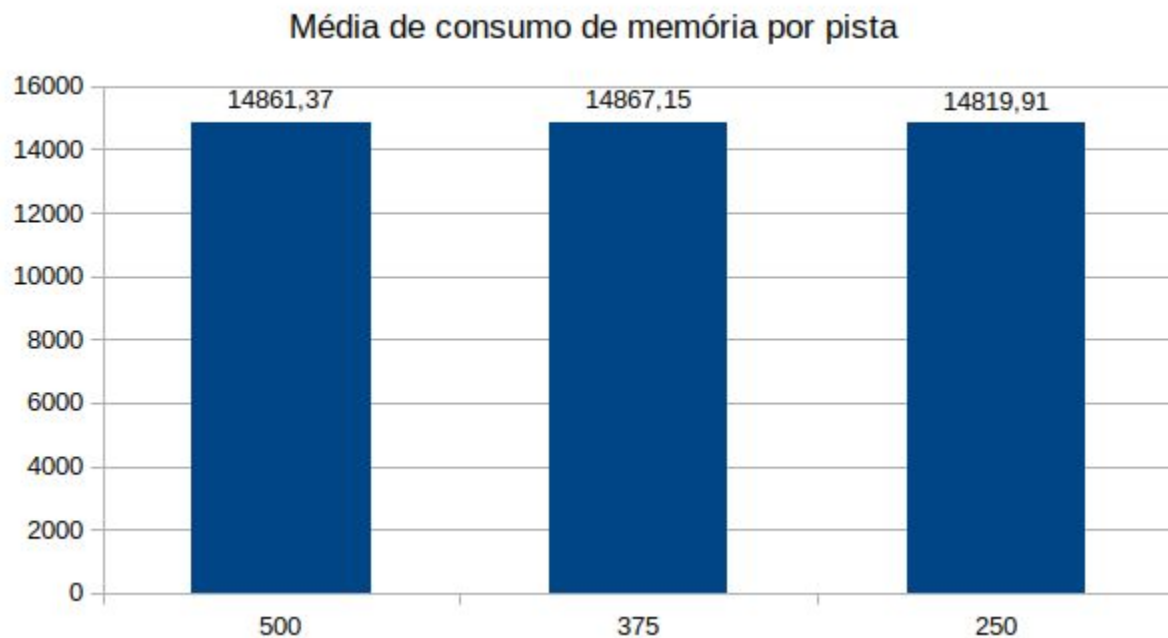
Experimentos

Para a realização de simulações, selecionamos as seguintes granularidade para tamanho da pista e quantidade de ciclistas:

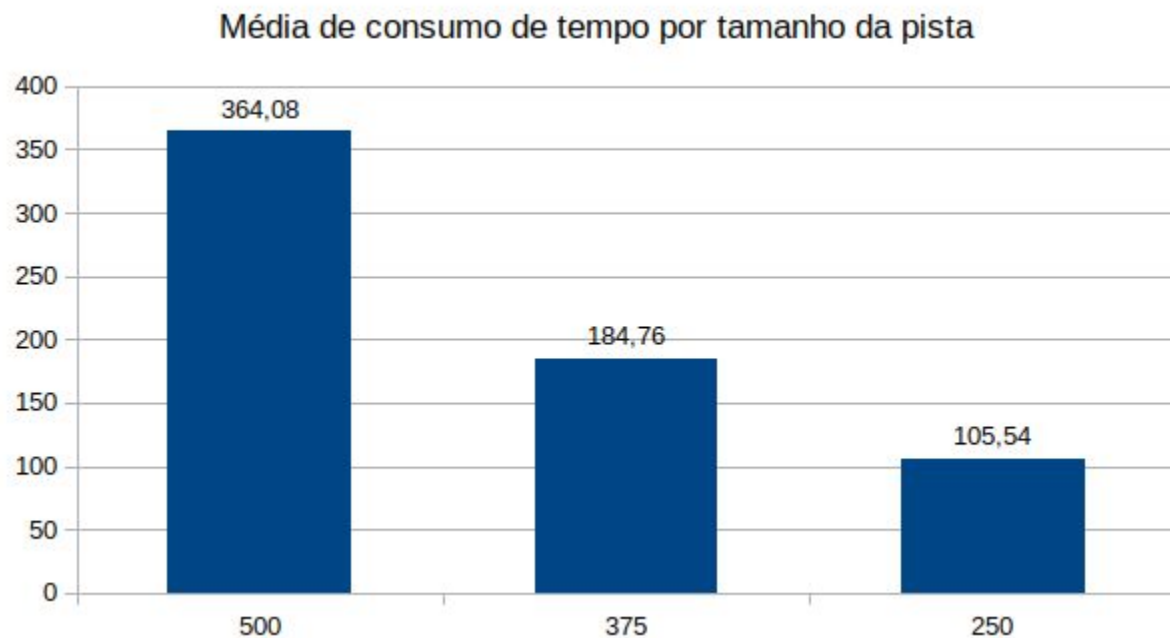
- 250, 375 e 500 ciclistas
- 10, 10 e 1000 ciclistas

Para cada configuração de pista realizamos 30 repetições, reduzindo duas casas decimais do tempo padrão de simulação do programa. Desta forma, ao invés de iterações a cada 60ms ou 20ms, são feitas iterações a cada 0,6ms e 0,2ms.

Resultados



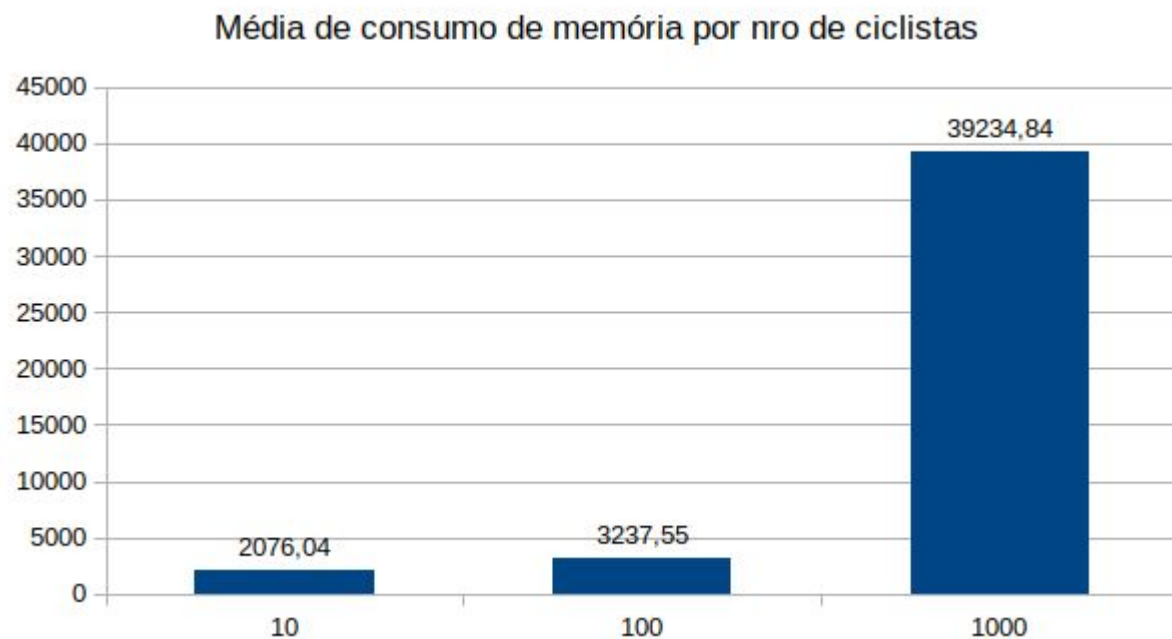
Resultados



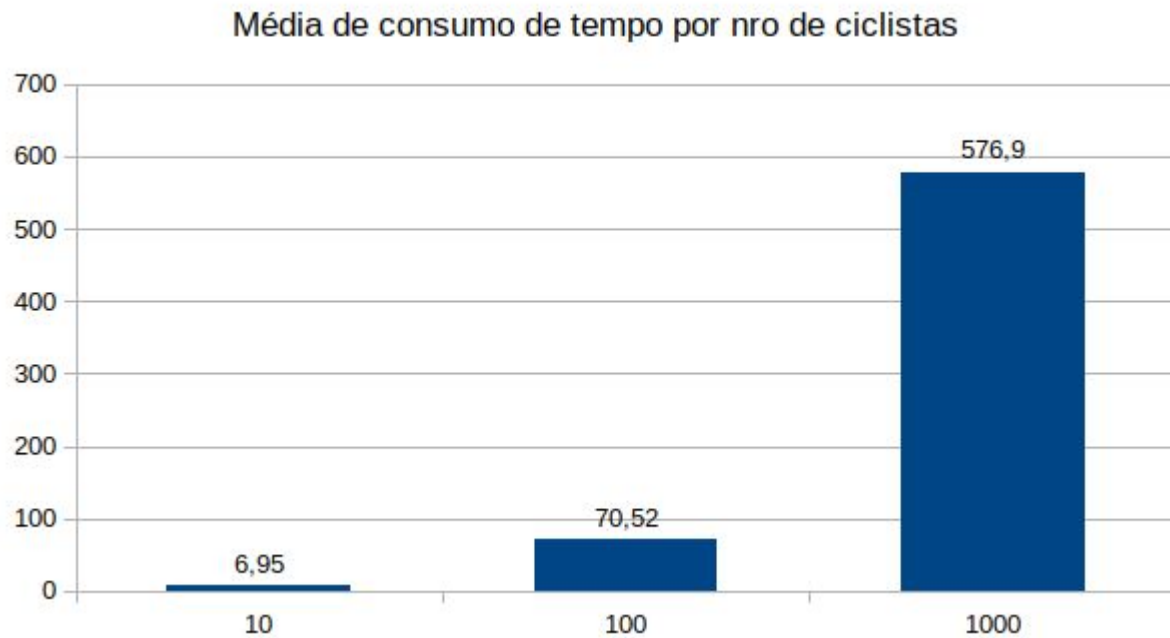
Resultados

Dados-Memória	Média	Intervalo de confiança
500p-1000c	39233,73333	117,6286431
500p-100c	3258,4	20,85004286
500p-10c	2092	20,27766429
375p-1000c	39283,33333	63,25400449
375p-100c	3235,6	24,8131932
375p-10c	2082,533333	19,71763747
250p-1000c	39187,46667	99,61878018
250p-100c	3218,666667	26,3514028
250p-10c	2053,6	19,08443011

Resultados



Resultados



Resultados

Dados-Tempo	Média	Intervalo de confiança
500p-1000c	978,4964291	12,68441555
500p-100c	104,6185475	2,551983831
500p-10c	9,126093767	0,2355812189
375p-1000c	473,7937914	36,67163609
375p-100c	73,4650346	4,80744598
375p-10c	7,009141667	0,1627840015
250p-1000c	278,4114933	10,04412472
250p-100c	33,4842764	0,7246216392
250p-10c	4,722630833	0,1029497521

Conclusões

A partir da análise do ***gráfico de memória para diferentes tamanhos de pista*** (utilizando o número médio de ciclistas: 100), é possível observar que a quantidade de memória utilizada em diferentes tamanhos de pista não se altera muito para o mesmo número de ciclistas analisado. Quanto maior a pista, era esperado que ocorresse maior consumo de memória. Não era esperado que a média de consumo de memória fosse maior para a pista de tamanho médio, mas isso veio a ocorrer.

Já o ***gráfico de tempo para diferentes tamanhos de pista*** (utilizando o número médio de ciclistas: 100) mostrou que quanto maior a pista, maior o tempo depreendido para que o programa termine. Isso já era esperado: quanto maior o tamanho da pista, mais tempo os ciclistas levam para terminar a volta.

Conclusões

O **gráfico de tempo para diferentes quantidades de ciclistas** (utilizando o estudo na pista média: 375) demonstra o que era esperado, ou seja, a quantidade de tempo utilizada para as quantidades pequena, média e grande, difere numa proporção próxima ou menor que 10. Ou seja, como a proporção de ciclistas testados é de 10 : 100 : 1000, e quanto mais ciclistas, mais possíveis quebras, além de que a eliminação de ciclistas sempre ocorre a cada 2 rodadas, a proporção de tempo gasto se dá na ordem do número de ciclistas (assim como da pista), podendo ser menor por conta da quebra de threads.

Conclusões

O **gráfico de tempo para diferentes quantidades de ciclistas** (utilizando o estudo na pista média: 375) demonstra o que era esperado, ou seja, a quantidade de tempo utilizada para as quantidades pequena, média e grande, difere numa proporção próxima ou menor que 10. Ou seja, como a proporção de ciclistas testados é de 10 : 100 : 1000, e quanto mais ciclistas, mais possíveis quebras, além de que a eliminação de ciclistas sempre ocorre a cada 2 rodadas, a proporção de tempo gasto se dá na ordem do número de ciclistas (assim como da pista), podendo ser menor por conta da quebra de threads.

Os **intervalos de confiança** observados são consistentes em ordem de grandeza para a memória utilizada, estando na casa de 10% ou menor em relação a sua média. Nas análises de tempo a variação é maior nos experimentos com 1000 ciclistas e consistente com 100 ou 10. Acreditamos ser por maiores interferências de outros processos rodando em conjunto por mais tempo.

Obrigada!