**Lista de Sistemas Distribuídos e Programação Concorrente**

Prof. MSc. Rodrigo D. Malara

OpenMP - Baixe os arquivos do Google Classroom para realizar os exercícios

1. omp\_bug1.c
   * Apresenta erros no momento da compilação.
   * Compile e execute o programa.
   * Verifique e resolva o problema para que o programa execute corretamente

Envie o código corrigido e um screenshot da execução do programa

1. omp\_bug2.c
   * Cada thread deverá gerar um valor próprio para a variável total
   * Compile e execute o programa.
   * Verifique e resolva o problema para que o programa execute corretamente

Envie o código corrigido e um screenshot da execução do programa

1. omp\_bug5.c
   * Usando SECTIONS, dois threads inicializam o seu próprio vetor e somam no vetor do outro thread mas um deadlock pode ocorrer.
   * Compile e execute o programa.
   * Verifique e resolva o problema para que o programa execute corretamente

Envie o código corrigido e um screenshot da execução do programa

## Respostas dos bug fix

omp\_bugfix1.c

EXPLICAÇÃO: Este exemplo tenta mostrar o uso da diretiva PARALLEL for-DO combinada. Ele falha porque o loop não vem imediatamente após a diretiva. As correções incluem a remoção de todas as instruções entre a diretiva PARALLEL for-DO e o loop real.

omp\_bugfix2.c

EXPLICAÇÃO: Os bugs neste caso são causados ​​por negligenciar o escopo das variáveis ​​TID e TOTAL como PRIVATE. Por padrão, a maioria das variáveis ​​OpenMP tem escopo como SHARED. Essas variáveis ​​precisam ser exclusivas para cada thread.

omp\_bugfix5.c

EXPLICAÇÃO: O problema em omp\_bug5 é que o primeiro thread adquire o locka e depois tenta obter o lockb antes de liberar o locka. Enquanto isso, o segundo thread adquiriu lockb e depois tenta obter locka antes de liberar lockb. A solução supera o impasse usando bloqueios corretamente.