

## Programação Paralela

### Sistemas de Informação – 2022.1 – Prof.º Philippe Leal

#### Quarto Trabalho Computacional - T4

Desenvolva um algoritmo, fazendo uso das estratégias implementadas anteriormente (Threads e Mecanismos de Controle de Concorrência), para solucionar um **Sistema Linear Triangular Superior**  $Ux = b$ , onde  $U_{n \times n}$  é uma matriz **Triangular Superior**.

O pseudocódigo para solucionar um Sistema Triangular Superior é demonstrado no Slide 05 da apresentação, que segue anexa a este arquivo.

Seu algoritmo tem que utilizar duas Threads: uma para calcular o valor de  $x$  (ou seja, solucionar o Sistema) e outra para calcular o somatório.

Pode ser utilizado o Sistema que consta no Slide 02 da apresentação para testar seu algoritmo. **Mas lembre-se que seu trabalho pode ser testado com um Sistema Triangular Superior diferente.**

#### Atenção

**Obs. 1:** Não é permitido utilizar os seguintes recursos:

- Criar duas Threads, matar estas duas Threads, depois criar mais duas Threads, matar estas duas Threads, e assim sucessivamente;
- Utilizar qualquer estratégia para simular a utilização do Controle de Concorrência a fim de garantir a ordem de execução das Threads, como listas, filas, pilhas, etc. O Controle de Concorrência tem que ser o disponibilizado na linguagem de programação do grupo para este fim.

**Obs. 2:** Caso o grupo esteja na dúvida se é possível ou não utilizar determinada estratégia, deve-se perguntar ao Professor nas aulas síncronas anteriores a aula de apresentação do trabalho.

## Importante

- O grupo deve enviar o trabalho (o código do algoritmo) até às **23h59** do dia **26/10/2022** para o seguinte e-mail:

[philippeal@yahoo.com.br](mailto:philippeal@yahoo.com.br)

- No dia da apresentação (**27/10/2022**), o grupo deverá apresentar o trabalho para o professor, onde o mesmo fará perguntas para o grupo (ou para um componente do grupo) sobre o algoritmo.

- Apenas um e-mail por grupo deve ser enviado. Isto é, não há a necessidade de cada componente do grupo enviar. No entanto, recomenda-se enviar uma cópia do e-mail para os demais componentes.

- O **ASSUNTO** do e-mail deve ter a seguinte formatação:

**BSI-PP-T4-PrimeiroNomeDosAlunos**

Exemplo de e-mail do grupo (fictício) formado pelos alunos Lucas Pereira, Renato da Silva e Miguel dos Santos que farão o trabalho na Linguagem C:

Para: philippeal@yahoo.com.br

De: Lucas Pereira

**Assunto:** BSI-PP-T4-Lucas-Renato-Miguel

**Anexo:** BSI-PP-T4-Lucas-Renato-Miguel.c

**Repare que o nome do arquivo (.c) deve ter a mesma formatação do “Assunto” do e-mail:**

BSI-PP-T4-Lucas-Renato-Miguel.c

- E-mail com o “**Assunto**” fora do padrão **NÃO SERÁ ACEITO**.

- A primeira linha de cada código deve conter: os nomes dos **componentes do grupo**.

- Utilize nomes sugestivos para as variáveis. Faça corretamente a indentação e comentários no código para facilitar seu entendimento. Estes itens serão avaliados.

- Preste atenção se está enviando a versão correta do trabalho, visto que será considerada a data da entrega da versão correta.

- Não deixe para enviar o trabalho na última hora, pois podem acontecer problemas com o envio.
- Caso o grupo esteja com dúvidas, tire-as com o professor **DURANTE A AULA SÍNCRONA VOLTADA PARA ESTE FIM**. Não deixe para a última hora.
- Os trabalhos serão avaliados utilizando o Sistema Operacional **Linux Mint 19.03 com gcc 7.5.0**.
- Trabalhos entregues com atraso receberão uma penalidade ***P***:

$$P = 0,9 - 0,2(d - 1)$$

$$\text{Nota Máxima} = \text{Nota do Trabalho} \times P$$

onde  $d$  ( $d \geq 1$ ) é o número de dias em atraso. Exemplo: se o grupo entregar com 1 dia de atraso, ele poderá ficar com, no máximo, 90% da nota do trabalho. Repare que após 5 dias de atraso não é mais possível entregar o trabalho. O atraso é contado em número de dias e não de horas. Caso o grupo envie o trabalho a qualquer hora do dia seguinte a data de entrega, será considerado 1 dia de atraso, e assim por diante.