

SISTEMAS OPERACIONAIS I

Prof. Dr. Caetano Mazzoni Ranieri

PROJETO DA DISCIPLINA

- Desenvolver uma aplicação concorrente que simule o comportamento de uma rede de entregas, em que encomendas são transportadas por veículos de um ponto de redistribuição até outro.
- A sincronização deve ser feita com uso de semáforos e variáveis de trava.
- Os trabalhos serão realizados em duplas ou, excepcionalmente, em grupos de 3 alunos.

ENUNCIADO

- Para fins de simplificação, considere que os pontos de redistribuição estão organizados sequencialmente.
- Na sua implementação, suponha que:
 - Há **S** pontos de redistribuição.
 - Há **C** veículos que representam os meios de transporte.
 - Há **P** encomendas a serem entregues.
 - Há **A** espaços de carga em cada veículo (todas as encomendas ocupam exatamente 1 espaço de carga).
 - Deve-se assegurar que $P \gg A \gg C$.
- O programa deve admitir argumentos de entrada determinando S, C, P e A ao iniciar a aplicação. O thread principal deverá receber esses argumentos e gerar um thread para cada uma das P encomendas, cada um dos C veículos e cada um dos S pontos de redistribuição especificados (numerá-los sequencialmente para identificação).
- Os veículos podem partir de pontos distintos e aleatórios, definidos na inicialização dos seus threads. A rede de distribuição admite que apenas um veículo seja atendido em um ponto por vez. Se um ponto estiver vazio (sem encomendas aguardando despacho), o veículo segue para o próximo ponto de redistribuição.
- As encomendas, ao chegarem a um ponto de redistribuição, ficam organizadas em uma fila controlada pelo ponto de redistribuição. Ao chegar no destino, cada encomenda é descarregada (tempo aleatório) e seu thread finaliza.

- O tempo de viagem entre um ponto de redistribuição e outro é **aleatório e não fixo** para todos os veículos. Um veículo pode ultrapassar outro veículo durante a viagem e chegar antes ao próximo ponto de redistribuição.
- Enquanto houver encomendas para serem entregues, os veículos continuam circulando entre os pontos. Quando um veículo chega ao último ponto, ele é direcionado novamente ao primeiro ponto (uma fila circular).
- Argumentos de entrada fornecidos aos threads das encomendas indicam os seus pontos de origem e destino. Esses argumentos são determinados aleatoriamente quando da criação dos threads de cada encomenda.
- Quando todas as encomendas tiverem sido entregues, os veículos param de circular e a aplicação acaba.
- As saídas do programa deverão ser:
 1. Uma tela para monitoramento em tempo real das encomendas, pontos de redistribuição e veículos.
 2. Os arquivos de rastro das encomendas gravados em disco.
 - a. Cada encomenda gera um arquivo de rastro contendo o número da encomenda, seus pontos de origem e destino, o horário que chegou ao ponto de origem, o horário que foi carregada no veículo, o identificador desse veículo e o horário em que foi descarregada no ponto de destino.

ENTREGA

- Código-fonte, preferencialmente no Github ou repositório semelhante (será considerado o último commit antes do prazo de entrega).
 - Incluir readme detalhando a implementação realizada.
- Apresentação em sala de aula de cada trabalho.
- **Prazo:** 25/11 (data da apresentação).