

Transcrição das perguntas feitas online
Consulta: apenas as folhas de API fornecidas
Duração e cotações foram as indicadas no teste online

Parte I

1. No contexto de prevenção de deadlocks, seria mais viável agir a nível da condição detenção e espera ou da condição recursos não preemptíveis? Explique as razões. A resposta inclui dizer o que é que fazia em cada uma.
2. Considere um sistema com a estratégia de evitamento de deadlocks estudada nas aulas. Tem 4 recursos (A, B, C, D) e encontra-se a correr 4 processos (P1, P2, P3, P4). O sistema encontra-se agora no instante t, descrito pelas informações constantes na figura.

Disponibilidade inicial

A	B	C	D
5	7	6	7

Procura máxima inicial

	A	B	C	D
P1	3	4	2	2
P2	1	2	3	2
P3	2	4	2	4
P4	2	1	3	1

Atribuição instante t

	A	B	C	D
P1	1	2	1	2
P2	1	1	2	1
P3	1	2	1	2
P4	2	1	1	0

3. A DLL `exame.dll` tem uma variável inteira `NumVal`. Pretende-se que os vários processos que usam a DLL possam usar essa variável `NumVal`, mas partilhando o seu valor, ou seja, se um processo lá colocar um valor, os outros vêm esse valor. Explique como se pode conseguir esse efeito usando DLL. Se entender que não é possível, explique porquê. Esta questão não envolve mecanismos de comunicação.
4. A função `trataJanela` atende os eventos de várias janelas. Pretende-se associar a cada uma dessas janelas uma cópia independente de um conjunto de dados de forma a que o atendimento de um evento possa trabalhar sobre os dados associados à janela onde ocorreu esse evento. Os dados em questão são o conjunto composto por `Nome` (10 caracteres), `morada` (10 caracteres), `telefone` (inteiro). Explique como pode fazer essa associação sem recorrer a variáveis globais. Não dê uma resposta superficial.
5. Considere um programa `win32` com uma janela. Pretende-se construir um jogo de estímulo de atenção nessa janela. Ao carregar na tecla espaço, aparecerá um carácter sorteado numa posição aleatória que muda de 1 em 1 segundo. A letra permanece visível durante 6 segundos ou até o utilizador carregar na tecla correspondente à letra que está na janela. Se o utilizador tiver acertado na tecla, a pontuação é incrementada. Assuma que existe esta função `int obtemValAleat(int min, int máximo)`. Escreva a função da janela para cumprir o pretendido. Se houver mais código diretamente envolvido nesta funcionalidade, apresente-o também. Não interessa apresentar aqui a função `main` nem aspectos genéricos tais como a criação da janela, mas pode haver outras inicializações envolvidas que sejam relevantes. O código deve ser contextualizado (“estas linhas pertencem à função...”, “esta variável etc”, este mecanismo X etc).

Parte II
Perguntas 1) a 3.b)

Nota prévia: Implemente SÓ E APENAS o que é pedido em cada questão. Código não relacionado com os aspetos pedidos em cada pergunta é ignorado para efeitos da avaliação.

Considere um sistema de marcação de viagens de avião composto por uma agência e vários balcões. A agência é uma aplicação consola que apenas tem uma única instância em execução. O balcão é uma aplicação gráfica e pode ter em execução múltiplas instâncias, mas a agência apenas consegue lidar com um máximo de 10 balcões em simultâneo. Toda a troca de informação entre estas aplicações é feita exclusivamente por memória partilhada. Pode usar outros mecanismos do sistema operativo para outros fins que não a troca de informação.

A marcação de viagens é feita em dois passos:

1. No primeiro passo, o balcão interage com a agência para se dar a conhecer e para obter um número maior ou igual a zero que o identificará posteriormente. Não podem existir dois balcões ativos com o mesmo número atribuído. Pode tirar partido deste número em todos os aspetos que achar útil (por exemplo, para uso em nomes de mecanismos utilizados entre processos). Caso a agência esteja na sua capacidade máxima, esta responde com um valor -123 e o balcão desiste, informando o utilizador. No lado da agência, todas as comunicações associadas a este primeiro passo são atendidas obrigatoriamente pela mesma *thread*.
 2. No segundo passo, o balcão envia o ID do voo pretendido (15 caracteres) e o número (inteiro) do lugar para a agência. Esta verifica se o lugar está disponível através da função `int verificaSePode(TCHAR *, int)`, que já se encontra implementada e pronta a utilizar. Esta função devolve 0 (recusa) ou 1 (aceita) com base em critérios que não são conhecidos, e demora uma quantidade de tempo variável (que pode ser grande). Finalmente, a resposta é dada ao balcão através da memória partilhada. Se o lugar lhe for dado considera-se a comunicação entre o balcão e a agência como terminada. Se o lugar tiver sido recusado, o balcão é livre de tentar novamente com outro lugar, ou desistir enviando para a agência o texto “desistir” no campo do ID do voo, o que leva ao término da comunicação. Em ambos os cenários, o balcão deve apresentar uma frase na sua janela principal indicando o resultado. Quando a comunicação com um balcão termina, a agência fica livre para reutilizar recursos que este estivesse a ocupar, como, por exemplo, o número identificativo. Neste segundo passo, do lado da agência, cada balcão é atendido por uma *thread* diferente (ou seja, existe uma por cada balcão).
-
1. Apresente uma proposta de sincronização para o sistema descrito. A sua resposta deve incluir uma descrição textual dos mecanismos necessários, onde e como são criados e utilizados. Apresente ainda as estruturas de dados que achar necessárias para a comunicação entre os processos.
 2. Ao nível da aplicação agência, escreva o código para as seguintes partes:
 - a) Implemente a *thread* responsável por atender os novos balcões no primeiro passo. Inclua na sua resposta o lançamento desta *thread*, contextualizando numa frase onde deve ocorrer.
 - b) Implemente a *thread* responsável por atender os pedidos de um dos balcões no segundo passo. Inclua na sua resposta o lançamento desta *thread*, contextualizado com a alínea anterior.

3. Ao nível da aplicação balcão, escreva o código para as seguintes partes:
- a) Implemente o código necessário para que a aplicação leia os campos ID do voo e número do lugar de uma *Dialog Box*, enviando de seguida o pedido de reserva com essa informação à agência. Assuma que a *Dialog Box* tem o ID **IDD_DIALOG_DADOS** e que as caixas de texto dos dois campos referidos têm os IDs **IDC_EDIT_VOO** e **IDC_EDIT_LUGAR**. Deve igualmente apresentar o código necessário para o lançamento da *Dialog Box*.
 - b) Implemente a *thread* que recebe a resposta da agência a uma reserva, e todo o código necessário para que seja apresentada no fundo da janela uma frase indicando a aceitação ou recusa dessa reserva. As frases que deve escrever são “Reserva de voo feita com sucesso.” e “Reserva de voo recusada pela agência.”.