

Roteiro de Estudos 6

Sincronização - Espera Ocupada

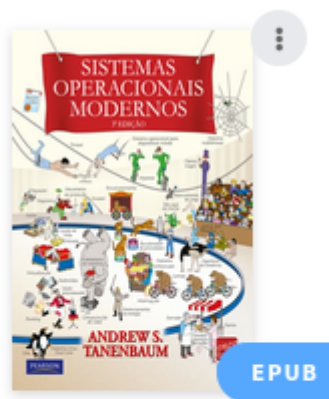
- **LEITURA**

- A. S. Tanenbaum "Sistemas Operacionais: projeto e implementação", 3a. Edição, Editora Pearson Prentice Hall, 2009.

- Seção 2.3 “Comunicação entre Processos” (até a sub-seção 2.3.4 inclusa) (pp. 70-77)

- **Para acessar o livro no acervo digital da UFES:**

1. Acesse <https://bibliotecas-digitais.ufes.br/> e entre com suas credenciais de login único ufes (não é o email... é só o login!)
2. Clique no botão “Acessar” da “**Biblioteca Virtual Pearson**”
3. Na nova aba/janela que abriu, digite na barra de busca “sistemas operacionais modernos”
4. Vão aparecer 4 resultados, selecione este:



★★★★★ (25)

Sistemas
Operacionais
Modernos - 3...

Tanenbaum, Andrew S.

- Observações no texto
 - “espera ociosa” == espera ocupada == busy wait

- **VÍDEO**

- [Aula da UNIVESP sobre Busy Wait](#)

- **RESUMÃO**

- [Slides com uma compilação do conteúdo](#)

- **EXERCÍCIOS (valendo turings!!)**

- **Responda o formulário fornecido juntamente com este roteiro**

=====

Lista de Exercícios de Consolidação

O objetivo da lista é ajudar no estudo individual dos alunos. Soluções de questões específicas poderão ser discutidas em sala de aula, conforme interesse dos alunos.

=====

1. Qual o problema com a solução que desabilita as interrupções para implementar a exclusão mútua?
2. O que é espera ocupada (busy wait) e qual o seu problema? Em que situações esse tipo de solução pode ser utilizado?
3. Em muitos sistemas operacionais existe uma chamada de sistema “yield()”, cujo efeito é passar o processo (ou thread) chamador para o final da fila de aptos (prontos). Suponha que um dado kernel escalone o processador (**há apenas 1!**) utilizando fatias de tempo e suporte a chamada “yield()”. Suponha ainda que, em todas as aplicações, o tempo necessário para executar uma seção (região) crítica seja menor que a duração de uma fatia de tempo. Para este caso em particular, buscando resolver o problema da seção crítica, considere o seguinte mecanismo de sincronização:
Imediatamente antes de entrar na seção crítica, a thread chama “yield()”;
Ao sair da seção crítica, a thread não faz nada.
Responda as questões abaixo, sempre justificando sua resposta.
 - (a) Esta solução oferece exclusão mútua?
 - (b) Esta solução apresenta a possibilidade de postergação indefinida (**starvation**)?
 - (c) Esta solução funciona para mais de duas threads/processos?
 - (d) Esta solução apresenta busy waiting?
4. Considerando a afirmação : “A técnica de busy waiting (spin lock) não deve ser empregada para sincronização em máquinas monoprocessadores, seu uso é indicado apenas em sistemas que executam sobre máquinas multiprocessadoras” Responda : Essa afirmação está correta ? Justifique a sua resposta.
5. No projeto de um sistema operacional para um monoprocessador a equipe de desenvolvimento utilizou um escalonador não preemptivo e a técnica de espera ocupada (busy waiting) na implementação de suas primitivas de exclusão mútua (lock/unlock). Pergunta-se : a decisão do tipo de escalonador e da técnica de busy waiting foi correta ?

Em que aspectos sua resposta seria modificada se esse sistema executasse em máquinas multiprocessadoras ? Justifique sua resposta.