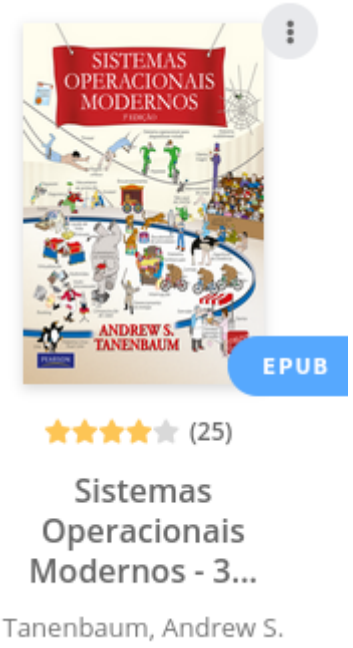
Roteiro de Estudos 8
Sincronização - Monitores

1. LEITURA

- A. S. Tanenbaum "Sistemas Operacionais: projeto e implementação", 3a. Edição, Editora Pearson Prentice Hall, 2009.
  - Seção 2.3.7 “Monitores” (pp. 81-85)
    - **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:



  - Observações no texto
    - “variável condicional” == variável de condição

2. RESUMÃO SOBRE MONITORES

- Slides com uma compilação do conteúdo

3. **VÍDEOS**
   - Introduction to Monitor in Process Synchronization Operating System (23'20")
   - Monitor Solution for Bounded Buffer Problem : Process Synchronization Concepts in Operating System (9'35")

4. **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. Considere a seguinte modelagem, por monitor, para o problema do produtor-consumidor com *n* produtores e *m* consumidores.

```
monitor buffer {
   int itens=0; cond temItens, temEspacos;
   ...
   int pega() {
      while (1) {
         if (!itens) wait(temItens);
         pega item no buffer
         itens--;
         signal(temEspacos);
         return (item);
      }
   }

   void coloca() {
      while (1) {
         if (itens==MAX) wait(temEspacos);
         coloca item no buffer
         itens++;
         signal(temItens);
      }
   }
}
```

Suponha que esse monitor funcione com a disciplina “sinaliza e continua" (abordagem de Hansen), dada em sala (mas observe que ele admite filas separadas por variáveis de condição). Explique por que essa solução não funciona corretamente.

2. Um porto marítimo de uma cidade pode receber no máximo N navios que entram através de um canal. Pelo canal pode transitar apenas um navio de cada vez. Os navios que saem do porto têm prioridade na travessia do canal. Cada navio é simulado por uma tarefa cujo pseudo-código é o seguinte:

```
thr_navio() {
< Chegada à entrada do canal >
pedido_entrada();
< Atravessa o canal >
entrada_OK();
< Atraca no porto >
pedido_saida();
< Atravessa o canal >
saida_OK();
< Parte em viagem para outro porto >
}
```

   Utilizando monitores, implemente em pseudo-código C as funções `pedido_entrada()`, `entrada_OK()`, `pedido_saida()` e `saida_OK()`.

   R: Uma possivel solução!

3. Utilizando as primitivas de semáforos, explique como implementar monitores seguindo a abordagem de Hoare e de Hansen.

4. (Problema do Barbeiro Dorminhoco) Uma barbearia tem um salão de corte com uma cadeira de barbeiro e uma sala de espera com 5 cadeiras. Os clientes entram na sala de espera, um por vez, se houver espaço disponível (isto é, se há alguma cadeira livre); se não, vão embora procurar outra barbearia. Toda vez que o barbeiro termina o corte de um cabelo, o cliente sai e vai a outra loja, e um (apenas um por vez) cliente que está sentado esperando (se houver algum) entra na sala de corte e corta seu cabelo. Se o barbeiro vir que a sala de espera está vazia, tira uma soneca no salão mesmo. Quando um cliente chega e vê que o barbeiro está dormindo, o barbeiro é acordado por esse cliente e corta o seu cabelo. Use um monitor para coordenar a operação do barbeiro e dos clientes.