

Roteiro de Estudos 10

Threads

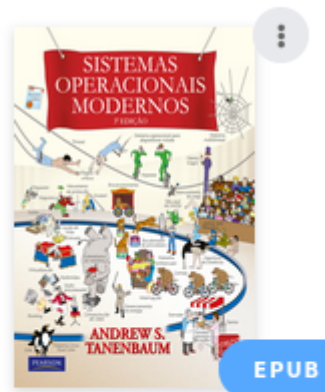
1. LEITURA

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

- Seção 2.2 "Threads" (até a seção 2.2.6 inclusa - pp. 57-67)

- **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.

2. RESUMÃO sobre Threads

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

3. VÍDEOS

- Teoria (Xoviabcs)

- [Threads: Thread Model, Thread vs Process, pthread library](#) (36'41")
 - Após 25min ele explica a biblioteca PTHREAD
- [Threads: User level & Kernel level thread. Threading issues](#) (15'57')

- Prática (CodeVault)

- [Short introduction to threads \(pthreads\)](#) (8'37")
- [Difference between processes and threads](#) (10'29")
- [What are Race Conditions?](#) (13'08")
- [What is a mutex in C? \(pthread_mutex\)](#) (9'17")
- [How to create threads in a loop \(pthread_create\)](#) (8'49")
- [How to pass arguments to threads in C](#) (13'51")
- [Practical example for using threads #1 \(Summing numbers from an array\)](#) (9'41")

4. EXERCÍCIOS (valendo turings!!)

- Atividade prática descrita no Classroom

=====

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. Explique a diferença entre unidade de alocação de recursos e unidade de escalonamento?
2. Quais as vantagens e desvantagens do compartilhamento do espaço de endereçamento entre threads de um mesmo processo?
3. Quais as diferenças entre threads em nível de usuário e threads em nível de kernel? Em quais circunstâncias um tipo é melhor do que o outro?
4. Quais estruturas são utilizadas para a criação de uma thread de kernel? Como elas se diferenciam daquelas utilizadas para a criação de um processo?

5. Como o uso de threads de kernel pode melhorar o desempenho de aplicações paralelas em ambientes com múltiplos processadores?
6. Joãozinho está desenvolvendo um sistema operacional com suporte a KLT. Mas ele se encontra com a seguinte dúvida: quando uma thread de um processo faz uma chamada de sistema `fork()`, o que deve acontecer? Pense em duas possibilidades diferentes de implementação e explique para Joãozinho a diferença entre elas.

- What happens when a thread invokes `fork`?
 - Duplicate all threads?
 - Not easily done... other threads may be running or blocked in a system call or a critical section
 - Duplicate only the caller thread?
 - More feasible.

7. Se uma thread causa um *Segmentation Fault*, apenas essa thread deveria ser terminada ou o processo inteiro? Faça o no Linux, utilizando a biblioteca `pthread`.