



# Conectividade em Sistemas Ciberfísicos

## Lista de Exercícios I

Exercícios sobre Introdução a S.O, Processos e Threads

# Exercício 1: Relacione corretamente as definições dos mecanismos de gerenciamento de recursos do S.O.

( ) Cria um modelo de abstração para escrita e leitura em dispositivos HDD (Hard Disk Drive) e SSD (Solid State Drive).

( ) Evita que um programa escreva sobre as variáveis de outro programa.

( ) Determina em qual core de um processador um programa será executado.

( ) Gerencia a escrita e leitura de dispositivos manipulados diretamente pelo usuário.

( ) Permite que mais de um programa seja executado simultaneamente no mesmo core de um processador.

( ) Utiliza a memória de armazenamento em massa no HDD ou SSD como uma extensão da memória física em RAM.

1. Gerencia de Processos
2. Gerencia de Memória
3. Gerencia de E/S
4. Gerencia de Armazenamento
5. n.d.a

## Exercício 2. Indique as afirmações corretas sobre o gerenciamento de processos

- I. Processos que efetuam operações de E/S, como leitura em disco, são colocados em estado de espera até que a operação seja completada.
- II. A recepção de pacotes pela rede, através de chamadas de sockets, são exemplos de operação que colocam processos em estado de espera.
- III. Processos que realizam muitas operações de E/S consomem muito tempo de CPU.
- IV. Processos relacionados a programas de inteligência artificial, como treinamento de redes neurais, realizam poucas operações de E/S e por isso consome muito tempo de CPU enquanto estão ativos.
- V. Programas que estão em estado de espera não consome CPU, mas eles consomem memória até que sejam terminados.

### Exercício 3. Indique as afirmações corretas sobre a diferença entre escalonamento cooperativo e preemptivo

- I. O algoritmo que determina em um dado momento qual processo irá ganhar tempo de CPU é denominado **algoritmo de escalonamento**
- II. O termo **troca de contexto** refere-se a mudança de estado de uma CPU quando o processo que está sendo executado é substituído por outro.
- III. No escalonamento **cooperativo** (ou não preemptivo), a troca de contexto acontece apenas quando um processo executa uma operação de E/S.
- IV. No escalomanto **preemptivo**, a troca de contexto ocorre através de interrupções, que limitam o tempo que um processo pode ocupar tempo de CPU.
- V. O escalonamento **cooperativo** pode ser **mais eficiente** que o escalonamento preemptivo, uma vez que ocorrem menos trocas de contexto.

## Exercício 4. Indique as diferenças entre processos e threads.

( ) Pode ser executado em cores diferentes de um processador.

( ) Compartilham o mesmo espaço de memória.

( ) Podem compartilhar variáveis em um mesmo programa.

( ) Permite que funções de um mesmo programa sejam executadas de forma paralela.

( ) Pode ser executado em computadores diferentes.

( ) Permite que a troca de contexto seja executada de forma relativamente rápida.

1. Processo
2. Thread
3. Ambos
4. Nenhum dos dois

## Exercício 5. Considerando as características dos processos e threads relacione as colunas.

- |  |                     |
|--|---------------------|
| <input type="checkbox"/> Desenvolver um servidor TCP onde o número de clientes conectados a cada instante é muito variável.  | 1. Apenas Threads   |
| <input type="checkbox"/> Desenvolver um programa com um número fixo de componentes (funções) que são executados de forma paralela.   | 2. Melhor Threads   |
| <input type="checkbox"/> Desenvolver um programa com um número fixo de componentes (funções) que são executados de forma paralela, e a falha de um componente não pode afetar os demais. | 3. Penas Processos  |
| <input type="checkbox"/> Utilizar os recursos de um processador com múltiplos cores.   | 4. Apenas Processos |
| <input type="checkbox"/> Desenvolver um programa cujos componentes são executados de forma paralela em computadores diferentes.  | 5. Indiferente      |
|  | 6. Nenhum dos dois  |