

# Fundamentos de Sistemas Operacionais

**Professor: Cristiano Bonato Both**



JESUÍTAS BRASIL



Somos infinitas possibilidades

# Atividade 05 - prática

1. Refaça o exercício apresentado em aula, modificando para tornar as threads em loop e descreva os comportamentos considerando os comandos:
  1. `ps -eLf`
  2. `Top` ----> `shift + H`
  3. `/proc`



# Atividade 05 - teórica

A correta utilização de processos e *threads* é fundamental para garantir o desempenho e a transparência de sistemas distribuídos. Sobre esse tema, considere as afirmativas a seguir.

I. A sobreposição de *threads* em um processo é o principal recurso para obtenção de alto grau de transparência de distribuição em redes com longos tempos de propagação de mensagens.

II. A desvantagem de se estruturar um programa para utilizar múltiplas *threads* é que ele ficará dependente de sistemas multiprocessadores.

III. O modelo de *threads* implementado pelo sistema operacional deve ser aquele em que o gerenciamento de *threads* fica inteiramente no espaço de cada processo para evitar trocas de contexto entre processos e o núcleo (*kernel*) no chaveamento de *threads*.

IV. Servidores *multithreaded* têm melhor desempenho se estruturados com ao menos uma *thread* despachante e várias *threads* operárias para recebimento e processamento de requisições.

Assinale a alternativa correta.

- a) Somente as afirmativas I e II são corretas.
- b) Somente as afirmativas I e IV são corretas.
- c) Somente as afirmativas III e IV são corretas.
- d) Somente as afirmativas I, II e III são corretas.
- e) Somente as afirmativas II, III e IV são corretas.



JESUÍTAS BRASIL



Somos infinitas possibilidades

# Atividade 05 - teórica

Considere o seguinte programa com dois processos concorrentes. O escalonador poderá alternar entre um e outro, isto é, eles poderão ser intercalados durante sua execução. As variáveis x e y são compartilhadas pelos dois processos e inicializadas antes de sua execução.

```
programa P
int x = 0;
int y = 0;
processo A {
    while (x == 0);
    print('a');
    y = 1;
    y = 0;
    print('d');
    y = 1;
}
```

```
processo B {
    print('b');
    x = 1;
    while (y == 0);
    print("c");
}
```

As possíveis saídas são:

- (a) adbc ou bcad
- (b) badc ou bacd
- (c) abdc ou abcd
- (d) dbca ou dcab
- (e) Nenhuma das opções anteriores.

