

- 1) Esboce o código de um programa que implemente uma comunicação entre processos pai e filho de forma que o pai determine a preempção do filho, ou seja, o pai sistematicamente (a cada 2 segundos) interrompe a execução do filho por um determinado tempo (outros 2 segundos) e em seguida libera a continuidade de sua execução. O processo filho exibe, a cada segundo, mensagem com o tempo de execução do programa em segundos.
- 2) Explique o que é STARVATION. Dê um exemplo
- 3) Diferencie os escalonamentos preemptivos e não-preemptivos. Indique os diferentes tipos de preempção.
- 4) Dentre as seguintes ações, quais **necessariamente** são executadas em modo kernel/supervisor? Apresente uma breve **justificativa para cada item**:
  - a) Aumentar/diminuir a área de memória usada pela pilha de execução (stack).
  - b) Verificar o valor de um semáforo quando há uma chamada a down().
  - c) Transferir dados entre processos através de memória compartilhada
  - d) Desabilitar e habilitar interrupções.
  - e) Registrar a chegada de um sinal para um processo
  - f) Aumentar ou diminuir a prioridade de um processo.
- 5) Indique se a afirmativa é Verdadeira ou Falsa, justificando em qualquer caso:
  - a) A MULTIPROGRAMAÇÃO, para ser implementada, exige MULTIPROCESSAMENTO.
  - b) A operação DOWN sobre um semáforo de valor nulo acarreta mudança de estado do processo, de EXECUTANDO para PRONTO para executar.
  - c) Sistemas operacionais "batch", assim como os de tempo real, são não-preemptivos, já que não utilizam "time-slice".
  - d) A exclusão mútua permite que dois processos acessem a mesma região crítica ao mesmo tempo, se ambos não estão alterando a mesma variável.