Infraestrutura de Software 2022.1 - 2o EE

1. Indique se as afirmações são verdadeiras (V) ou falsas (F) e justifique cada uma das respostas. (2,0)

(i) Sistemas distribuídos baseados na arquitetura Peer-to-Peer são mais velozes do que aqueles baseados na arquitetura Cliente-Servidor;

(ii) A comunicação em sistemas multiprocessadores é baseada em compartilhamento de memória;

(iii) Sistemas que não conseguem estabelecer previsibilidade nas execuções não são apropriados para tempo real crítico;

(iv) Sistemas de tempo real são baseados em computação rápida.

2. Explique o envolvimento do S.O. na ocorrência de uma falta de página (2,0).

3. Explique a diferença de tabela de alocação de arquivos e i-nodes. (1,5)

4. Explique através de exemplos os algoritmos de substituição de página *working set* e *menos frequentemente usada.* (2,0)

5. **Implemente um sistema** ferroviário usando pthreads, no qual cada trem é representado por uma *thread*. Um trem precisa obter 20 passageiros na estação para seguir viagem e, quando não há passageiros na estação, ficam ociosos (“dormindo”) esperando por alguma notificação de novos passageiros. Quando conseguem obter 20 passageiros, o trem segue viagem e conclui sua execução. O trem só consegue pegar um passageiro por vez, e esta situação é representada por um incremento de um contador local a *thread*. A estação é representada por uma *thread* que tem capacidade máxima de 5 passageiros. Quando a estação está vazia, ela preenche todas as 5 entradas com passageiros. A quantidade de passageiros na estação é representada por um contador global (contador\_estacao) para todas as *threads.* Quando a estação fica cheia (contador\_estacao == 5), a *thread* estação dorme. Se alguma entrada da estação fica vazia (contador\_estacao < 5), a thread deve ser notificada e, em seguida, preenche o espaço vazio com um novo passageiro (incremetando o contador contador\_estacao). Ao chegar um novo passageiro, a estação deve notificar cada trem. Assuma o mesmo código para as *threads* representado os trens. **Existe condição de disputa neste sistema? Explique brevemente.**

**Obs: Não precisa mostrar a criação das threads, mas você deve demonstrar a utilização de outros recursos necessários para a resposta. (2,5).**

int pthread\_mutex\_unlock(pthread\_mutex\_t \*mutex); int pthread\_mutex\_lock(pthread\_mutex\_t \*mutex); int pthread\_cond\_wait(pthread\_cond\_t \*cond, pthread\_mutex\_t \*mutex); int pthread\_cond\_signal(pthread\_cond\_t \*cond);int pthread\_cond\_broadcast(pthread\_cond\_t \**cond*); int pthread\_mutex\_init(pthread\_mutex\_t \*mutex, const pthread\_mutexattr\_t \*attr);int pthread\_mutex\_destroy(pthread\_mutex\_t \*mutex); int pthread\_join(pthread\_t *thread*, void \*\**value\_ptr*)

.