1. Explique e exemplifique a razão de técnicas de escalonamento necessitarem incluir restrições temporais para escalonar tarefas em sistemas de tempo real. (2,0).

2. Em sistemas de arquivos, o que seria superbloco? Ademais, como o espaço livre em memória secundária (ex: HDD, SSD) pode ser gerenciado? (2,0)

3. No contexto de memória virtual, o que seria a tabela de páginas? Explique o uso da tabela pelo sistema operacional e o hardware. (2,0)

4. Explique a definição de sistemas multiprocessadores, multicomputador e sistemas distribuídos (1,5)

5. Implemente um sistema de gerenciamento de bagagens de um aeroporto. A aplicação é composta por *threads* que representam várias esteiras as quais executam em um laço infinito. Após receber uma bagagem na esteira (chamada à função *receberBagagem())* , a esteira deverá ficar parada esperando que outras esteiras recebam também uma bagagem. Por exemplo, se há 4 esteiras, uma esteira que receber uma bagagem primeiro, deverá ficar bloqueada esperando o recebimento de uma bagagem pelas outras 3 esteiras, e assim sucessivamente. Após todas as esteiras receberem uma mala, o sistema de coleta é executado, e as esteiras voltam a funcionar para recebimento de uma nova bagagem. Assuma que a função *void receberBagagem()* recebe uma bagagem e você não precisa se preocupar com a implementação (basta chamá-la). Ademais, não precisa implementar o sistema de coleta, mas deve focar na sincronização das esteiras. **Além da implementação, explique brevemente se exclusão mútua é uma técnica apropriada para esse problema.** (2,5)

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

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*); int pthread\_barrier\_wait(pthread\_barrier\_t \*barrier); int pthread\_barrier\_init(pthread\_barrier\_t \* barrier, const pthread\_barrierattr\_t \* attr, unsigned int count );