**Infraestrutura de Software - 1º EE (2019.2)**

1. O que é contexto do processo e troca de contexto? O custo de troca de contexto do processo é maior do que uma *thread*? Explique. (2,0)

2. *Deadlock* é uma situação que pode ocorrer quando múltiplos processos ou threads estão presentes e cooperando em um sistema. Quais as condições necessárias para ocorrer um *deadlock* e explique 2 formas de previni-las (2,0)

3. Na implementação, o *kernel* de um sistema operacional pode ser estruturado de diferentes formas. Descreva 3 formas distintas de estruturação do *kernel*  (2,0).

4. Faça resumo de como funciona e a inicialização do *bootloader* de 2 etapasem um arquitetura x86*.* Quais são os cuidados necessários no código ASM?(ex: localização disco, memória...). (1,5)

5. **Implemente um sistema** de exibição de voos em um aeroporto, no qual é composto por várias threads. Cada thread fica em um laço infinito enviando dados para uma tela usando a função exibirTela (char \*texto). Esta função não realiza controle de acesso, mas acessos simultâneos podem afetar a exibição correta do voo na tela. Assuma o mesmo código para todas as threads e não se preocupe com o código da função exibirTela. **Qual nome da condição existente no sistema? Explique.** .

**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\_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*)