**Infraestrutura de Software - 2º Chamada/Final (2018.1)**

1. O que seria *middleware?* Por que há diferentes categorias de *middleware?* Descreva brevemente 2 tipos de *middleware.* (2,0).

2 Em sistemas de arquivos, o que seria o superbloco? (1,5).

3. Façam um comparativo entre processos e *threads* na programação concorrente. Descreva vantagens e desvantagens (1,5)

4. Implemente famoso jogo “pedra-papel-tesoura” usando threads. O programa deverá ter **T** threads, no qual cada um representará um jogador. Como entrada,  o usuário deverá informar quantidade de jogadores **T** e a quantidade de rodadas **N**. A saída consiste em apresentar uma mensagem mostrando o placar final. O jogo será automático, havendo somente a interação com o usuário no início.

Obs: Abstraia o código das threads e mostre sincronização. (3,0)

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

5. A figura abaixo apresenta um código envolvendo 2 threads e 3 regiões críticas. Assuma o uso de 3 mutex, um para cada região crítica (ex: mutex1 -> regiao\_critica1()). Assim, responda: (i)As quais as 4 condições necessárias para ocorrência de *deadlock* estã presentes? Explique. (ii) Caso positivo, demonstre uma solução (2,0)

