

**Sistemas de Informação - EACH-USP**  
**Sistemas Operacionais - Prof. Alexandre da Silva Freire**  
**Prova Substitutiva - 7 de dezembro de 2016**

Nome: \_\_\_\_\_ NUSP: \_\_\_\_\_ Nota: \_\_\_\_\_

**ATENÇÃO ÀS SEGUINTE INSTRUÇÕES:** (1) Pode fazer a prova a lápis; (2) Escreva seu nome completo e número USP em todas as folhas, inclusive nesta; (3) Devolva todas as folhas, inclusive os rascunhos e esta folha; (4) Desligue todos os aparelhos eletrônicos e não converse durante a prova; (5) Pode fazer as questões em qualquer ordem; (6) Faça apenas o que as questões pedem, sem se preocupar com o restante do código; (7) Qualquer dúvida, levante a mão e aguarde o professor chegar até você ou vá até o professor (não fale em voz alta).

**Questão 1 (Valor: 2.5).** Para que serve o *Algoritmo do Grafo de Alocação de Recursos* (AGAR)? Descreva, detalhadamente, como ele funciona. Para que serve o *Algoritmo do Banqueiro*? Descreva, detalhadamente, como ele funciona.

**Questão 2 (Valor: 2.5).** Descreva, de forma detalhada, como funciona a criptografia de chave assimétrica. Descreva um contexto no qual a mensagem deve ser cifrada utilizando uma chave pública e um outro contexto no qual a mensagem deve ser cifrada utilizando chave privada. Justifique seus exemplos. Explique como funcionam as “assinaturas digitais” e os “certificados digitais”.

**Questão 3 (Valor: 2.5).** Desenhe o *Diagrama de Gantt* para ilustrar o funcionamento do escalonador *Round-Robin* para a seguinte situação:

Processo	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>
Duração do próximo <i>burst</i> de CPU	8	9	15	2	11

Assuma que o *quantum* de tempo definido é de 3 unidades de tempo. Qual é o tempo médio de espera obtido? Há algum outro escalonador que obterá um tempo médio de espera inferior ao *Round-Robin* para a mesma situação descrita acima? Se sim, desenhe o *Diagrama de Gantt* correspondente e calcule o respectivo tempo médio de espera.

**Questão 4 (Valor: 2.5).** Escreva uma classe `Database` em Java, que contenha os métodos `void acquireReadLock()`, `void acquireWriteLock()`, `void releaseReadLock()` e `void releaseWriteLock()` para controlar o acesso a uma base de dados com múltiplos acessos concorrentes (ou seja, sua classe deve apresentar uma solução para o *Problema de Leitores e Escritores*). **Dica: utilize os métodos `void wait()` e `void notify()` do Java.**

*Boa prova!*