



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
CEARÁ

Departamento de Telemática
Avaliação de Software de Tempo-Real
Prof. Paulo Régis C. de Araújo

Nome: JOÃO GABRIEL CARNEIRO MEDEIROS

01. Cite um exemplo de um sistema que deve apresentar alta disponibilidade e um exemplo de um sistema que deve oferecer alta confiabilidade. (4 scores)
02. Em um programa, 20 processos concorrentes devem utilizar 8 blocos de memória de tamanho fixo. Um programa gerenciador de memória deve alocar esses blocos para os processos concorrentes. Em relação à interação, como podemos classificar esses processos? (2 scores)
03. Represente em (linguagem C) o funcionamento de um bloco de recuperação. Indique no código onde está o teste de aceitação, o ponto de recuperação, o módulo principal e os módulos alternativos. (6 scores)
04. Implemente um (programa concorrente) que permita que os processos A, B e C preencham um vetor de 10 elementos na seguinte ordem: O processo A deve preencher o primeiro elemento com a letra A. Posteriormente, o processo B deve preencher o segundo elemento do vetor com a letra B. E só depois, o processo C deve preencher o terceiro elemento com a letra C. O procedimento se repete com o processo A preenchendo a letra no quarto elemento. Todo o procedimento se repete até o vetor ser completamente preenchido. (8 scores)

04) No: João Gabriel Carneiro Medeiros.

1) * Alta Disponibilidade: Sistema Bancário, Aéreo,
* Alta Confiabilidade: Sistema de Tráfego, Aéreo.

02) Podemos classificá-los como competidores!

Isso, pois, note que o FATO DA AGÃO DE ALOCAÇÃO DOS RECURSOS REQUER DE FORMA INEVITÁVEL A CO-SOL DO SISTEMA, E O QUE ACABA, PORTANTO, OS CLASSIFICANDO COMO PROCESSOS COMPETIDORES.

03) VISTA O "PSEUDO-CÓDIGO" ABAIXO:

Bloco de Recuperação {

int x = 0;

int indice = 0;

Do { } → "Ponto de Recuperação"

If (indice == 0) {

x = F1(); } → "Mód. Principal."

If (indice == 1) {

x = F2(); } → "Mód. Alternativo"

If (indice == 2) {

x = F3(); } → "Mód. Alternativo"

} While [indice <= 5] && (x != 5) }
"Teste de Aceitação" →

}

05.) (....)

```
void *thread_A(void *threadno){  
    while (indice <= 7){  
        while (flagA == 0){  
            LETRAS[indice] = 'A';  
            indice++;  
            flagA = 0;  
            flagB = 1;  
        }  
        return null;  
    }  
}
```

```
void *thread_B(void *threadno){  
    while (indice <= 8){  
        while (flagB == 0){  
            LETRAS[indice] = 'B';  
            indice++;  
            flagB = 0;  
            flagC = 1;  
        }  
        return null;  
    }  
}
```

```
void *thread_C(void *threadno){  
    while (indice <= 9){  
        while (flagC == 0){  
            LETRAS[indice] = 'C';  
            indice++;  
            flagC = 0;  
            flagA = 1;  
        }  
        return null;  
    }  
}
```