

## **QUESTÕES DO LIVRO**

**Real-Time Systems, Jane Liu**

**Cap 6: 4,5,8,9,10,11,12,15,16,21**  
**fazer diagramas, usar análise por cálculo do tempo de resposta**

**Cap 8: 1,2,7 fazer diagramas**

**Conceitos Básicos e Técnicas de Implementação de Sistemas Automatizados**  
**2001/1 – Prova P2 – Prof. Rômulo**

Nome:

-----

1) (2 pontos) Que tipos de critérios devem ser utilizados no momento da definição da fatia de tempo a ser empregada em um determinado sistema ? Cite dois.

2) (2 pontos) Qual a fragmentação apresentada pelos métodos de gerência de memória baseados em partições fixas, partições variáveis, paginação simples e segmentação simples? Justifique as 8 situações.

3) (2 pontos) Descreva o erro na implementação do produtor-consumidor mostrada abaixo. Crie uma seqüência de eventos que termina em algum comportamento indesejado para o programa.

```
struct tipo_dado buffer[N];
int proxima_insercao = 0;
int proxima_remocao = 0;
...
semaphore exclusao_mutua = 1;
semaphore espera_vaga = N;
semaphore espera_dado = 0;
...
void produtor( void)
{
    ...
    P( exclusao_mutua );
    P( espera_vaga );
    buffer[ proxima_insercao ] = dado_produzido;
    proxima_insercao = ( proxima_insercao + 1 ) % N;
    V( espera_dado );
    V( exclusao_mutua );
    ...
}
...
void consumidor( void)
{
    ...
    P( espera_dado );
    P( exclusao_mutua );
    dado_a_consumir = buffer[ proxima_remocao ];
    proxima_remocao = ( proxima_remocao + 1 ) % N;
    V( exclusao_mutua );
    V( espera_vaga );
    ...
}
```

4) (2 pontos) Dadas as tarefas T1, T2 e T3 com as restrições temporais descritas abaixo, e sabendo que T1 precede T2, calcule os tempos de respostas das três tarefas. A atribuição de prioridades segue o deadline monotonic.

Tarefa	release jitter	tempo de computação	período	deadline
T1	1	9	80	30
T2	-	8	80	40
T3	-	7	50	50

5) (2 pontos) Suponha que o fornecedor de um Sistema Operacional de Tempo Real informou que o tempo para chaveamento de contexto é 100us e a latência máxima no atendimento a interrupções é de 500us. Cite mais duas informações que seriam necessárias para aplicar sobre o sistema as equações da teoria de escalonamento de tempo real vista em aula. Justifique a necessidade dessas duas informações. Cite apenas duas.

**Conceitos Básicos e Técnicas de Implementação de Sistemas Automatizados  
2002/1 – Prova Prof. Rômulo – segunda-feira**

Nome:

-----  
**1)(1.5 pontos)** O sistema operacional é um programa dirigido por eventos, e esses eventos são sinalizados por interrupções. Para cada uma das três classes de interrupções (periférico, proteção, chamada de sistema), descreva a reação que o sistema operacional deverá ter com relação aos estados dos processos (quais transições de estado são desencadeadas).

**2)(1.5 pontos)** Qual a fragmentação apresentada pelos métodos de gerência de memória baseados em partições fixas, partições variáveis e paginação simples? Explique as 6 situações.

**3)(2.0 pontos)** Considerando o código abaixo, uma variação da implementação do produtor/consumidor vista em aula, responda as seguintes perguntas (justificando):

- a) Está garantido o acesso exclusivo à variável "buffer" no caso de 1 produtor e 1 consumidor ?
- b) É possível a ocorrência de postergação indefinida no caso de 3 produtores e 1 consumidor ?

```
struct tipo_dado buffer;  
semaphore espera_vaga = 1;      semaphore espera_dado = 0;
```

```
...  
void produtor( void)  
{  
    ...  
    P( espera_vaga );  
    buffer = dado_produzido;  
    V( espera_dado );  
    ...  
}
```

```
void consumidor( void)  
{  
    ...  
    P( espera_dado );  
    dado_a_consumir = buffer;  
    V( espera_vaga );  
    ...  
}
```

**4)(1.5 pontos)** Usando os mecanismos "mutex" e "variáveis condição" do Posix, implemente as operações P e V de um semáforo. Cada uma dessas operações deverá ser substituída por um código C com semântica similar. Lembre-se que estas operações devem ser atômicas.

Dicas sobre o Posix:

```
pthread_mutex_t m = PTHREAD_MUTEX_INITIALIZER;  
pthread_mutex_lock( &m );  
pthread_mutex_unlock( &m );
```

```
pthread_cond_t vc = PTHREAD_COND_INITIALIZER;  
pthread_cond_wait( &vc , &m );  
pthread_cond_signal( &vc );
```

**5)(1.5 pontos)** Determine se o conjunto de tarefas abaixo é escalonável com deadline monotônico, aplicando a análise para  $D \leq P$ , vista em aula.

T1:	J1=2	C1=4	P1=10	D1=10	B1 = 0
T2:	J2=3	C2=5	P2=50	D2=30	B2 = 0
T3:	J3=0	C3=10	P3=60	D3=60	B3 = 0

**6)(2 pontos)** Suponha que o fornecedor de um Sistema Operacional de Tempo Real informou que o tempo para chaveamento de contexto é 100us e a latência máxima no atendimento a interrupções é de 500us. Cite mais duas informações que seriam necessárias para aplicar sobre o sistema as equações da teoria de escalonamento de tempo real vista em aula. Justifique a necessidade dessas duas informações. Cite apenas duas.

**Conceitos Básicos e Técnicas de Implementação de Sistemas Automatizados  
2002/1 – Prova Prof. Rômulo – sexta-feira**

Nome:

-----

**1)(1.5 pontos)** Que critérios devem ser utilizados no momento da definição da fatia de tempo a ser empregada em um determinado sistema ? Cite dois.

**2)(1.5 pontos)** Considere um sistema operacional que trabalha com paginação simples. As páginas são de 1Kbyte. O endereço lógico é formado por 16 bits. O endereço físico é formado por 20 bits. Qual o tamanho do:

- (a) Espaço de endereçamento lógico (maior programa possível)?
- (b) Espaço de endereçamento físico (memória principal)?
- (c) Entrada da tabela de páginas, sem considerar bits de proteção?
- (d) Tabela de páginas (número de entradas necessárias no pior caso)?

**3)(2.0 pontos)** Considerando o código abaixo, uma variação da implementação do produtor/consumidor vista em aula, responda as seguintes perguntas (justificando):

- a) Está garantido o acesso exclusivo à variável "buffer" no caso de 1 produtor e 1 consumidor ?
- b) É possível a ocorrência de postergação indefinida no caso de 3 produtores e 1 consumidor ?

```
struct tipo_dado buffer;
semaphore espera_vaga = 1;      semaphore espera_dado = 0;
...
void produtor( void)
{
    ...
    P( espera_vaga );
    buffer = dado_produzido;
    V( espera_dado );
    ...
}

void consumidor( void)
{
    ...
    P( espera_dado );
    dado_a_consumir = buffer;
    V( espera_vaga );
    ...
}
```

**4)(1.5 pontos)** Usando os mecanismos "mutex" e "variáveis condição" do Posix, implemente as operações P e V de um semáforo. Cada uma dessas operações deverá ser substituída por um código C com semântica similar. Lembre-se que estas operações devem ser atômicas.

Dicas sobre o Posix:

```
pthread_mutex_t m = PTHREAD_MUTEX_INITIALIZER;  
pthread_mutex_lock( &m );  
pthread_mutex_unlock( &m );
```

```
pthread_cond_t vc = PTHREAD_COND_INITIALIZER;  
pthread_cond_wait( &vc , &m );  
pthread_cond_signal( &vc );
```

**5)(1.5 pontos)** Determine se o conjunto de tarefas abaixo é escalonável com deadline monotônico, aplicando a análise para  $D \leq P$ , vista em aula.

T1:	J1=2	C1=4	P1=10	D1=10	B1 = 1
T2:	J2=2	C2=5	P2=50	D2=30	B2 = 0
T3:	J3=0	C3=10	P3=60	D3=60	B3 = 0

**6)(2 pontos)** Com respeito às equações usadas na questão anterior, explique de que maneira os seguintes tempos de um sistema operacional aparecerão nos cálculos:

- Tempo gasto na execução do tratador de interrupções do timer que ocorre enquanto uma tarefa da aplicação executa;

- Processo faz chamada de sistema X mas outro processo com prioridade menor fez a mesma chamada de sistema X antes dele e está usando de forma exclusiva as estruturas de dados associadas com a chamada de sistema X em questão.

**Conceitos Básicos e Técnicas de Implementação de Sistemas Automatizados 2003/1 – Rômulo**  
**Nome:**

-----

1)(2 pontos) Suponha que ocorreu uma interrupção de hardware causada por um periférico. Descreva o algoritmo seguido pelo sistema operacional na reação a esta interrupção, com respeito aos estados dos processos. Teste seu algoritmo supondo que ocorreu:

- (a) Uma interrupção do controlador de disco sinalizando a conclusão de uma operação.
- (b) Uma interrupção do temporizador em hardware sinalizando que passaram mais 10 ms.

2)(2 pontos) Considerando paginação e segmentação:

- (a) Qual a maior vantagem da paginação sobre a segmentação ?
- (b) Qual a maior vantagem da segmentação sobre a paginação ?

Cite as vantagens e também crie um exemplo para cada uma onde a vantagem citada se manifesta.

3)(2 pontos) Usando os mecanismos **mutex** e **variável condição** do Posix, implemente um monitor para o seguinte sistema: A tarefa “controle” executa um laço onde ela lê o sensor, armazena o valor lido, compara com o valor de referência fixo e envia um valor calculado para o atuador. Caso o valor lido seja maior que o limite, ela acorda a tarefa “alarme”. A tarefa “alarme” apenas espera ser acordada pelo sensor e escreve uma mensagem na tela que inclui o valor lido.

Dicas sobre o Posix:

```
pthread_mutex_t m = PTHREAD_MUTEX_INITIALIZER;  
pthread_mutex_lock( &m );  
pthread_mutex_unlock( &m );  
  
pthread_cond_t vc = PTHREAD_COND_INITIALIZER;  
pthread_cond_wait( &vc , &m );  
pthread_cond_signal( &vc );
```

4)(2 pontos) Determine se o conjunto de tarefas abaixo é escalonável com deadline monotônico, aplicando a análise para  $D \leq P$ , vista em aula.

T1:	J1=0	C1=2	P1=5	D1=5	B1 = 0
T2:	J2=0	C2=5	P2=30	D2=10	B2 = 0
T3:	J3=0	C3=10	P3=25	D3=25	B3 = 0

$$W_i = C_i + B_i + \sum_{j \in HP(i)} \left\lceil \frac{J_j + W_i}{P_j} \right\rceil \times C_j \quad R_i = J_i + W_i$$

5)(2 pontos) Ilustre através de exemplos como as características abaixo do sistema operacional podem aumentar o tempo de resposta de uma tarefa com alta prioridade. Para cada exemplo, desenhe um diagrama de tempo para mostrar o impacto sobre a tarefa de alta prioridade.

- (a) Tempo de chaveamento entre tarefas.
- (b) Kernel não é preemptivo.



**Conceitos Básicos e Técnicas de Implementação de Sistemas Automatizados 2004 – Rômulo**  
**Nome:**

-----

1)(1,5 ponto) Em um sistema usando paginação pura, o espaço de endereçamento lógico de cada processo consiste de, no máximo, 1 Mbyte de tamanho. As páginas físicas são de 512 bytes. Entretanto, a memória física pode ter até 32 Mbytes de tamanho. Diga quantos bits são necessários para especificar cada uma das grandezas abaixo, explicando de onde veio cada número:

- (a) Número da página lógica;
- (b) Número da página física;
- (c) Número de entradas máximo na tabela de páginas.

2)(1 ponto) Uma vez que paginação não apresenta fragmentação externa enquanto segmentação apresenta fragmentação externa, quais motivações existem para justificar a idéia de segmentação ? Quais as duas vantagens apontadas para a segmentação pura em relação a paginação pura ?

3)(2 pontos) Usando **semáforos**, implemente rotinas cujo comportamento seja similar ao das funções wait e signal vistas em aula e descritas abaixo. Crie tantos semáforos e variáveis auxiliares do tipo inteiro quantos julgar necessário. Explique a lógica da solução.

WAIT( VC ): Sempre bloqueia o processo até alguém executar um SIGNAL.

SIGNAL( VC ): Se existe alguém bloqueado em VC, acorda apenas 1 deles.

4)(2 pontos) Em um sistema que suporta programação concorrente apenas através da troca de mensagens, será criado um Servidor para controlar o envio de dados entre produtores e consumidores. Os dados serão enviados diretamente de um Produtor para um Consumidor livre, mas o Produtor deve consultar o Servidor para obter o número do Consumidor que ele deverá usar. O Servidor deve manter uma lista com os números dos Consumidores livres no momento, e informar um desses números quando um Produtor solicita. Após receber e processar os dados, o Consumidor volta a informar que está livre para o Servidor. Mostre o algoritmo do Servidor, em português estruturado. Apresentar a solução como foi feito para o Servidor de Buffer visto em aula. Supor “receive” bloqueante.

5)(2 pontos) Determine se o conjunto de tarefas abaixo é escalonável se prioridades forem atribuídas segundo o deadline monotônico, aplicando a análise para  $D \leq P$ , vista em aula. Considere que existe uma seção crítica entre T1 e T2, em função de uma tabela acessada pelas duas tarefas. A tarefa T1 permanece 1 unidade de tempo acessando a tabela, enquanto a tarefa T2 permanece 2 unidades de tempo acessando a mesma tabela. O protocolo herança de prioridade é utilizado.

T1:	J1=0	C1=2	P1=5	D1=5
T2:	J2=0	C2=5	P2=30	D2=10
T3:	J3=0	C3=10	P3=25	D3=25

$$W_i = C_i + B_i + \sum_{j \in HP(i)} \left\lceil \frac{J_j + W_i}{P_j} \right\rceil \times C_j \quad R_i = J_i + W_i$$

6)(1,5 ponto) Ilustre através de exemplos como as características abaixo do sistema operacional podem aumentar o tempo de resposta de uma tarefa com alta prioridade. Para cada exemplo, desenhe um diagrama de tempo para mostrar o impacto sobre uma tarefa de alta prioridade e procure indicar do que depende a quantidade de atraso sofrida.

- (c) Tempo de chaveamento entre tarefas;
- (d) Kernel não é preemptivo;
- (e) Tempo gasto com o tratador de interrupções do timer.

## Conceitos Básicos e Técnicas de Implementação de Sistemas Automatizados 2005 – Rômulo

Nome: \_\_\_\_\_

1)(2 pontos) Em um sistema usando paginação pura, o espaço de endereçamento lógico de cada processo consiste de, no máximo, 1 Mbyte de tamanho. As páginas físicas são de 512 bytes. Entretanto, a memória física pode ter até 32 Mbytes de tamanho. Diga quantos bits são necessários para especificar cada uma das grandezas abaixo, explicando de onde veio cada número:

- (d) Número da página lógica;
- (e) Número da página física;
- (f) Número de entradas máximo na tabela de páginas;
- (g) Deslocamento de um byte dentro da página.

2)(2 pontos) Considere a situação de uma estrada que possui uma ponte cuja largura só comporta um carro. A estrada como um todo possui uma pista em cada sentido, mas a ponte possui uma única pista. Carros vindos do norte e carros vindos do sul chegam a ponte com uma só pista e precisam atravessá-la. Os que viajam no mesmo sentido podem atravessar a ponte ao mesmo tempo, mas os que viajam em sentido oposto precisam esperar que a ponte fique livre.

Modele os carros como threads e a ponte como um recurso, o qual pode ser utilizado simultaneamente por carros que travegam no mesmo sentido, mas nunca por carros em sentidos opostos. Implemente uma solução para este problema usando mutexes e variáveis condição da biblioteca das pthreads. A solução consiste de 2 rotinas, como mostrado abaixo:

```
/* função chamada pela thread carro para solicitar o acesso a ponte, informando o seu sentido (NORTE ou SUL), ao retornar desta função a thread está autorizada a entrar na ponte */  
void pede_para_entrar (int sentido);
```

```
/* função chamada pela thread carro para avisar que saiu da ponte, possivelmente liberando processos do outro sentido */  
void avisa_que_saiu (int sentido);
```

### Dicas sobre o Posix:

```
pthread_mutex_t m = PTHREAD_MUTEX_INITIALIZER;  
pthread_mutex_lock( &m );  
pthread_mutex_unlock( &m );
```

```
pthread_cond_t vc = PTHREAD_COND_INITIALIZER;  
pthread_cond_wait( &vc , &m );  
pthread_cond_signal( &vc );
```

3)(2 pontos) Usando **semáforos**, implemente rotinas cujo comportamento seja similar ao das funções wait e signal vistas em aula e descritas abaixo. Crie tantos semáforos e variáveis auxiliares do tipo inteiro quantos julgar necessário. Comente o papel de cada variável criada.

WAIT( VC ): Sempre bloqueia o processo até alguém executar um SIGNAL.

SIGNAL( VC ): Se existe alguém bloqueado em VC, acorda apenas 1 deles.

4)(2 pontos) Em um sistema que suporta programação concorrente apenas através da troca de mensagens, será criado um Servidor para controlar o envio de dados entre produtores e consumidores. Os dados serão enviados diretamente de um Produtor para um Consumidor livre, mas o Produtor deve consultar o Servidor para obter o número do Consumidor que ele deverá usar. O Servidor deve manter uma lista com os números dos Consumidores livres no momento, e informar um desses números quando um Produtor solicita. Após receber e processar os dados, o Consumidor volta a informar que está livre para o Servidor. Mostre o algoritmo do Servidor, em português estruturado. Apresentar a solução como foi feito para o Servidor de Buffer visto em aula. Supor “receive” bloqueante.

5)(2 pontos) Determine se o conjunto de tarefas abaixo é escalonável se prioridades forem atribuídas segundo o deadline monotônico, aplicando a análise para  $D \leq P$ , vista em aula. Considere que existe uma seção crítica entre T1 e T2, em função de uma tabela acessada pelas duas tarefas. A tarefa T1 permanece 1 unidade de tempo acessando a tabela, enquanto a tarefa T2 permanece 2 unidades de tempo acessando a mesma tabela. O protocolo herança de prioridade é utilizado.

T1:	J1=0	C1=2	P1=5	D1=5
T2:	J2=0	C2=5	P2=30	D2=10
T3:	J3=0	C3=10	P3=25	D3=25

$$W_i = C_i + B_i + \sum_{j \in HP(i)} \left\lceil \frac{J_j + W_i}{P_j} \right\rceil \times C_j \quad R_i = J_i + W_i$$