

1. O que garante a atomicidade (indivisibilidade) da execução de uma instrução em um processador.
2. Quando um processo está executando instruções do programa do usuário, diz-se que o processo está no modo usuário. Quando este processo faz uma chamada de sistema, diz-se que o processo está no modo "kernel". Quais as diferenças de execução no modo usuário e no modo kernel? O que acontece na passagem do modo usuário para o modo kernel?
3. Em diversos sistemas operacionais o utilitário "ps" lista os processos do sistema. Que estruturas são necessárias para tal?  
Descreva em alto nível o algoritmo do "ps" com relação ao acesso a tais estruturas.
4. Se uma interrupção está sendo tratada pelo sistema e outra interrupção acontece por algum dispositivo, quais são as opções de projeto para o tratamento de ambas?
5. Considere um esquema de gerência de memória com partições fixas de mesmo tamanho de  $2^{16}$  bytes e memória total de  $2^{24}$  bytes. Para cada processo em memória o Sistema Operacional mantém uma referência a uma partição. Quantos bits são necessários para esta referência?
6. Um bloco de 1MByte é gerenciado com o sistema Buddy. Mostre em uma figura (arvore) o resultado da seguinte sequência de operações: a) request 70 Kbytes; b) request 35 Kbytes; c) request 80 Kbytes; retorna alocação de (a); d) request 60 Kbytes; retorna (b).
7. Defina as estruturas de dados que você utilizaria para manter o controle de partições livres (a) em um esquema de partições fixas e (b) em um esquema de partições variáveis.
8. Suponha que a tabela de páginas do processo executando no processador é como abaixo. Os valores são decimais, todos iniciando de zero, e endereços são de bytes de memória. O tamanho da página é 1.024 bytes.

Virtual page number	Valid bit	Reference bit	Modify bit	Page frame number
0	1	1	0	4
1	1	1	1	7
2	0	0	0	—
3	1	0	0	2
4	0	0	0	—
5	1	0	1	0

- a. Descreva como um endereço virtual gerado pela CPU é traduzido para um endereço físico.
  - b. A qual endereço físico (se válido) corresponderia cada um dos seguintes endereços virtuais (não trate page faults aqui).
    - i. 1.052
    - ii. 2.221
    - iii. 5.499
9. Em um sistema operacional com escalonamento Round-Robin, descreva as ações relativas a escalonamento que ocorrem a cada interrupção de relógio ao final de um quantum de tempo (ou time slice).
  10. Descreva como uma operação de leitura de arquivo "read(fileDescriptor,tam,&buffer)", com o arquivo a ser lido, tamanho e endereço do buffer de retorno, é realizada pelo sistema operacional. Aborde aspectos de chaveamento de contexto, trocas de estados de processos, e escalonamento.

11. Descreva o que acontece quando a leitura acima está pronta pelo dispositivo de I/O. Aborde a aspectos de chaveamento de contexto, escalonamento e transferência de valores para a área do processo.
12. Em um sistema operacional baseado em prioridades, preemptivo, descreva em que situações o escalonamento pode ocorrer e o que ocorre com os processos envolvidos.
13. Assuma um sistema em funcionamento, em que no tempo 5 não há recursos sendo utilizados, exceto processador e memória. O escalonamento ocorre conforme a política compatível com os eventos observados, sendo que a cada interrupção também ocorre escalonamento. Os seguintes eventos acontecem:
- Tempo 5: P1 executa um comando para ler do disco 3
  - Tempo 15: a fatia de tempo de P5 expira
  - Tempo 18: P7 executa um comando para escrever no disco 3
  - Tempo 20: P3 executa um comando para ler do disco 2
  - Tempo 24: P5 executa um comando para escrever no disco 3
  - Tempo 33: uma interrupção ocorre do disco 2. A leitura de P3 está pronta
  - Tempo 38: P8 termina
  - Tempo 40: interrupção do disco 3. Escrita de 5 está pronta
  - Tempo 44: P5 é swapped in
  - Tempo 48: uma interrupção ocorre do disco 3. A escrita de P7 está completa.
- Para os tempos 22, 37 e 47, identifique que estado cada processo está.  
Se um processo está bloqueado, identifique o evento pelo qual está bloqueado.

14. No grafo a seguir, quais arestas representam ações de escalonamento de:

- i. Longo
- ii. Médio e
- iii. Curto prazo

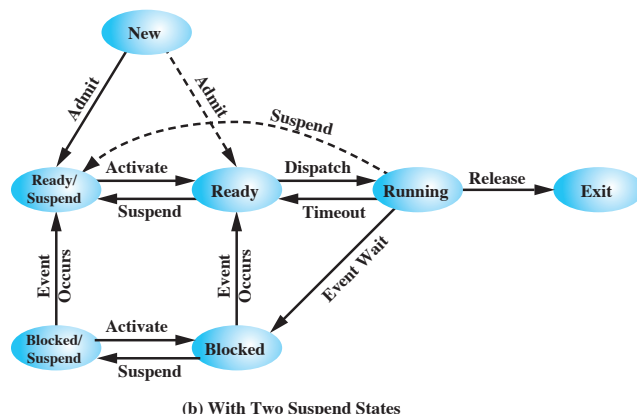


Figure 3.9 Process State Transition Diagram with Suspend States

15. Qual a diferença entre escalonamento shortest process next e shortest remaining time ?
16. Um sistema utiliza escalonamento HRRN. Toda vez que um processo é colocado na fila ready, existe uma avaliação para computar o processo com HRR. A computação da taxa de resposta é dada através da fórmula:

$$Ratio = \frac{time\ spent\ waiting + expected\ service\ time}{expected\ service\ time}$$

O expected service time é dado através da amostragem de burst times do processo. com a formula  $expectedTime = \alpha * ultimaAmostra + (1-\alpha) * expectedTime$ .  $\alpha = 0,8$ . O expectedTime default de início de um processo é 10. A partir do tempo  $t=0$ , o sistema tem dois processos com as seguintes amostras de burst time. Descreva que escalonamentos são realizados ao longo do tempo até que as amostras acabem.

P1: 5, 7, 4, 3, 8

P2: 8, 10, 7, 12, 9

17. Como se pode conceber um sistema concorrente livre de deadlock ?
18. Discuta se os seguintes algoritmos implementam seção crítica, discorrendo sobre:
- Exclusão Mútua
  - Progresso (não postergação)
  - Bloqueio

boolean wantp ← false, wantq ← false	
<b>p</b>	<b>q</b>
loop forever p1: non-critical section p2: await wantq = false p3: wantp ← true p4: critical section p5: wantp ← false	loop forever q1: non-critical section q2: await wantp = false q3: wantq ← true q4: critical section q5: wantq ← false

boolean wantp ← false, wantq ← false	
<b>p</b>	<b>q</b>
loop forever p1: non-critical section p2: wantp ← true p3: await wantq = false p4: critical section p5: wantp ← false	loop forever q1: non-critical section q2: wantq ← true q3: await wantp = false q4: critical section q5: wantq ← false

19. Porque um semáforo tem que ser implementado no sistema operacional ?
20. Dado a estrutura para o modelo produtor-consumidor com semáforos para controle de tamanho e exclusão mútua, abaixo. Complete a inicialização dos semáforos e suas utilizações nas threas Produtor e Consumidor.

```

N = 10
buffer tem tamanho N
semaforos
  mutex=
  cheio=
  vazio=

Produtor{
  loop{

    inserção circular
    elemento em buffer

  }
}

```

```

Consumidor{
  loop {

    retirada circular de
    elemento de buffer

  }
}

```