Atividade de sistemas operacionais
Capítulo 02

01° Explique o que é, para que serve e o que contém um PCB - Process Control Block.

Resposta: é uma estrutura de dados no núcleo do sistema operacional que serve para armazenar a informação necessária para tratar um determinado processo. O PCB contém informações críticas do processo ele deve ficar armazenado em uma área da memória protegida do acesso de usuários.

02° O que significa time sharing e qual a sua importância em um sistema operacional?

Resposta: Time-sharing é o termo para um tipo de multiprocessamento quando o sistema operacional implementa uma técnica de partilhar o uso da cpu por intervalo de tempo. Cada processo que recebe o controle tem um intervalo de tempo para processar e depois devolver o controle, que será repassado para outro processo e assim sucessivamente.

03° Como e com base em que critérios é escolhida a duração de um quantum de processamento?

Resposta: A duração atual do quantum depende do tipo de sistema operacional. No Linux por exemplo ela varia de 10 a 200 milissegundos, dependendo do tipo e prioridade da tarefa. Vários critérios podem ser definidos para a avaliação de escalonadores. Os mais frequentemente utilizados são:
Tempo de execução ou de vida" turnaround time": diz respeito ao tempo total da "vida" de cada tarefa, ou seja, o tempo decorrido entre a criação da tarefa e seu encerramento, computando todos os tempos de processamento e de espera.
Tempo de espera" waiting time": é o tempo total perdido pela tarefa na fila de tarefas prontas, aguardando o processador.
Tempo de resposta" response time": é o tempo decorrido entre a chegada de um evento ao sistema e o resultado imediato de seu processamento.

04° Considerando o diagrama de estados dos processos apresentado na figura a seguir, complete o diagrama com a transição de estado que está faltando (t6) e apresente o significado de cada um dos estados e transições.

Resposta:

05° Indique se cada uma das transições de estado de tarefas a seguir definidas é possível ou não. Se a transição for possível, dê um exemplo de situação na qual ela ocorre (N: Nova, P: pronta, E: executando, S: suspensa, T: terminada).

Resposta:

06° Relacione as afirmações abaixo aos respectivos estados no ciclo de vida das tarefas (N: Nova, P: Pronta, E: Executando, S: Suspensa, T: Terminada):

(N) O código da tarefa está sendo carregado.
(P) A tarefas são ordenadas por prioridades.
(E) A tarefa sai deste estado ao solicitar uma operação de entrada/saída.
(T) Os recursos usados pela tarefa são devolvidos ao sistema.

(P) A tarefa vai a este estado ao terminar seu quantum.

(E) A tarefa só precisa do processador para executar.
(S) O acesso a um semáforo em uso pode levar a tarefa a este estado.
(E) A tarefa pode criar novas tarefas.
(E) Há uma tarefa neste estado para cada processador do sistema.
(S) A tarefa aguarda a ocorrência de um evento externo.

07° Desenhe o diagrama de tempo da execução do código a seguir, informe qual a saída do programa na tela (com os valores de x) e calcule a duração aproximada de sua execução.

Resposta:

08° Indique quantas letras "X" serão impressas na tela pelo programa abaixo quando for executado com a seguinte linha de comando: a.out 4 3 2 1 Observações:
• a.out é o arquivo executável resultante da compilação do programa.
• A chamada de sistema fork cria um processo filho, clone do processo que a executou, retornando o valor zero no processo filho e um valor diferente de zero no processo pai.

Resposta:

09° O que são threads e para que servem?

Resposta: Thread é um pequeno programa que trabalha como um subsistema, sendo uma forma de um processo se auto dividir em duas ou mais tarefas. Os threads servem para se executar mais de um processo ao mesmo tempo.

10° Quais as principais vantagens e desvantagens de threads em relação a processos?

Resposta: Vantagens: Leve e de fácil implementação. Como o núcleo somente considera um thread, a carga de gerência imposta à divisão de recursos entre as tarefas. núcleo é pequena e não depende do número de threads dentro da aplicação.
Desvantagens: Como essas operações são intermediadas pelo núcleo, se um thread de Usuário solicitar uma operação de E/S o thread de núcleo correspondente será suspenso até a conclusão da operação, fazendo com que todos os threads de usuário associados ao processo parem de executar enquanto a operação não for concluída. Outro problema desse modelo diz respeito á divisão de recursos entre as tarefas.

11° Forneça dois exemplos de problemas cuja implementação multi-thread não tem desempenho melhor que a respectiva implementação sequencial.

Resposta: O modelo de threads "multi-thread" é adequado para a maioria das situações e atende bem às necessidades das aplicações interativas e servidores de rede. No entanto, é pouco escalável: a criação de um grande número de threads impõe uma carga significativa ao núcleo do sistema, inviabilizando aplicações com muitas tarefas como grandes servidores Web e simulações de grande porte.

12° Associe as afirmações a seguir aos seguintes modelos de threads: a) many-to-one (N:1) b) one-to-one (1:1); c) many-to-many (N:M):

(A) Tem a implementação mais simples, leve e eficiente.

(B) Multiplexa os threads de usuário em um pool de threads de núcleo.
(B) Pode impor uma carga muito pesada ao núcleo.
(A) Não permite explorar a presença de várias CPUs pelo mesmo processo.
(C) Permite uma maior concorrência sem impor muita carga ao núcleo.
(B) Geralmente implementado por bibliotecas.
(A) É o modelo implementado no Windows NT e seus sucessores.
(C) Se um thread bloquear, todos os demais têm de esperar por ele.
(C) Cada thread no nível do usuário tem sua correspondente dentro do núcleo.
(A) É o modelo com implementação mais complexa.

13° Considerando as implementações de threads N:1 e 1:1 para o trecho de código a seguir, a) desenhe os diagramas de execução, b) informe as durações aproximadas de execução e c) indique a saída do programa na tela. Considere a operação sleep () como uma chamada de sistema (syscall). Significado das operações:
• thread_create: cria um novo thread, pronta para executar.
• thread_join: espera o encerramento do thread informado como parâmetro.
• thread_exit: encerra a thread corrente.

Resposta:

14° Explique o que é escalonamento round-robin, dando um exemplo.

Resposta: Round-robin (RR) é um dos algoritmos mais simples de agendamento de processos em um sistema operacional, que atribui frações de tempo para cada processo em partes iguais e de forma circular, manipulando todos os processos sem prioridades. Escalonamento Round-Robin é simples e fácil de implementar. A adição da preempção por tempo ao escalonamento FCFS dá origem a outro algoritmo de escalonamento bastante popular, conhecido como escalonamento por revezamento, ou Round-Robin. FIFO (First in, first out) ou FCFS (First come, first served): Onde como seu próprio nome já diz, o primeiro que chega ser ao primeiro a ser executado; SJF (Shortest Job First): Onde o menor processo ganhará a CPU e atrás do mesmo formar uma fila de processos por ordem crescente de tempo de execução;

15° Considere um sistema de tempo compartilhado com valor de quantum tq e duração da troca de contexto ttc. Considere tarefas de entrada/saída que usam em média p% de seu quantum de tempo cada vez que recebem o processador. Defina a eficiência E do sistema como uma função dos parâmetros tq, ttc e p.

Resposta: $E = tq / tq + ttc$

16° Explique o que é, para que serve e como funciona a técnica de aging.

Resposta: Para evitar a inanição quando um processo nunca assegura um recurso e garantir a proporcionalidade expressa através das prioridades estáticas, um fator interno denominado envelhecimento “task aging” deve ser definido.
O envelhecimento indica há quanto tempo uma tarefa está aguardando o processador e aumenta sua prioridade proporcionalmente. Dessa forma, o envelhecimento evita a inanição dos processos de baixa prioridade, permitindo a eles obter o processador periodicamente.

17° A tabela a seguir representa um conjunto de tarefas prontas para utilizar um processador:
Indique a seqüência de execução das tarefas, o tempo médio de vida (tournaround time) e o tempo médio de espera (waiting time), para as políticas de escalonamento a seguir:
(a) FCFS cooperativa

(b) SJF cooperativa
(c) SJF preemptiva (SRTF)
(d) PRIO cooperativa
(e) PRIO preemptiva
(f) RR com tq = 2, sem envelhecimento
Considerações: todas as tarefas são orientadas a processamento; as trocas de contexto têm duração nula; em eventuais empates (idade, prioridade, duração, etc.), a tarefa ti com menor i prevalece; valores maiores de prioridade indicam maior prioridade. Para representar a sequência de execução das tarefas use o diagrama a seguir. Use × para indicar uma tarefa usando o processador, − para uma tarefa em espera na fila de prontos e para uma tarefa que ainda não iniciou ou já concluiu sua execução.

Resposta:

18° Idem, para as tarefas da tabela a seguir:

Resposta:

19° Explique os conceitos de inversão e herança de prioridade.

Resposta: A inversão de prioridades consiste em processos de alta prioridade serem impedidos de executar por causa de um processo de baixa prioridade. Um exemplo de como pode ocorrer uma invasão de prioridades:
1 Em um dado momento, o processador está livre e é alocado a um processo de baixa prioridade pb;
2 durante seu processamento, pb obtém o acesso exclusivo a um recurso e começa a usá-lo;
3 pb perde o processamento, pois um processo com prioridade maior que a dele (pm) foi acordado devido a uma interrupção.
4.pb volta ao final da fila de tarefas prontas, aguardando o processador; enquanto ele não voltar a executar, o recurso R permanece alocado a ele e ninguém poderá usá-lo;
5.Um processo de alta prioridade pa recebe o processador e solicita acesso ao recurso R; como o recurso está alocado ao processo pb, pa é suspenso até que o processo de baixa prioridade pb libere o recurso. Neste momento, o processo de alta prioridade pa não pode continuar sua execução, porque o recurso de que necessita está nas mãos do processo de baixa prioridade pb. Dessa forma, pa deve esperar que pb execute e libere R, o que justifica o nome inversão de prioridades.

20° Você deve analisar o software da sonda Mars Pathfinder discutido no livro-texto. O sistema usa escalonamento por prioridades preemptivo, sem envelhecimento e sem compartilhamento de tempo. Suponha que as tarefas tg e tm detêm a área de transferência de dados durante todo o período em que executam. Os dados de um trecho de execução das tarefas são indicados na tabela a seguir (observe que tg executa mais de uma vez).

Desenhe o diagrama de tempo da execução sem e com o protocolo de herança de prioridades e discuta sobre as diferenças observadas entre as duas execuções.

Resposta: