

Sistemas Operacionais

Processos e *threads*

Profa Ms. Adriane Ap. Loper

1

- Unidade de Ensino: 2
- Competência da Unidade: Processos e *threads*
- Resumo: Veremos como os processos e as *threads* funcionam e como trabalhamos com melhor potencial de um S.O.
- Palavras-chave: processos, *threads*, escalonamento
- Título da Teleaula: Processos e *threads*
- Teleaula nº: 2

2

Contextualização

O seu desafio é gerenciar rotinas e processos, criar, excluir e executar comandos para o processamento de dados de forma a otimizar o sistema operacional que interage com o ERP.

Se você tivesse que realizar esse serviço em uma clínica médica que acabou de adquirir um módulo de sistema integrado de gestão, como realizaria essa tarefa sem afetar as múltiplas operações em andamento?

3

Contextualização

Resolveremos todos esses requisitos conhecendo Sistemas Operacionais!

Vamos entender o que esse software nos permite fazer?

Iniciamos agora um novo e muito interessante aprendizado!

4

Conceitos

Introdução a processos

5

Estrutura de um sistema operacional

Inicialização do Sistema Operacional

- ✓ Programa carregador de inicialização(*bootstrap*) é executado quando computador é ligado (ROM);
- ✓ Realiza testes de hardware;
- ✓ Carrega o kernel do SO, do disco rígido (1º bloco de bytes) para a memória principal (RAM);
- ✓ A máquina utiliza um programa chamado BIOS armazenado em um CHIP ROM;

6

System Call

Para cada uma das rotinas que o sistema executará, há um mecanismo de controle de chamadas de sistema, o *system call*, que pode ser explícito ou implícito.

O *system call* é responsável por verificar os parâmetros da solicitação e enviar a sua respectiva resposta com o estado do processo, no caso, concluído, ou se houve algum erro e precise retornar à pilha de processos. Para cada serviço existe um *system call* associada e cada sistema operacional tem o seu próprio conjunto de chamadas.

7

Modos de tratamento de dados em processos

Hardware	Software	Armazenamento
Os dados do processo ficam armazenados nos registradores (status, PC e SP).	Há a especificação de recursos e suas limitações para que possam ser alocados os processos. Nome, PID, Owner (usr), prioridade, data/ hora de criação, tempo de processador, quotas e privilégios.	Refere-se à área de memória que será alocado o processo para que possa ser executado.

Fonte: Adaptado de Machado e Maia (2013, p. 4)

8

Contexto

Em contexto de *hardware*, os dados são tratados de acordo com o seu estado de processamento e armazenado no respectivo registro responsável por armazenar aquela determinada informação.

No contexto de *software*, as informações que o S.O. deve controlar referem-se à quantidade de arquivos que poderão ser abertos concomitantemente e quais processos detêm prioridade de execução, tamanho do *buffer* de E/S. Considera a PID e Owner - identificação do processo e de quem o criou - quotas (limites de recursos alocados) e privilégios.

No contexto do armazenamento, cada processo possui um endereço específico na memória.

9

Processo

A ideia principal é que um processo constitui uma atividade. Ele possui programa, entrada, saída e um estado. Um único processador pode ser compartilhado entre os vários processos, com algum algoritmo de escalonamento usado para determinar quando parar o trabalho sobre um processo e servir outro.

Os processos são divididos em estados: execução (*running*), pronto (*ready*) e espera (*wait*).



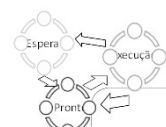
Fonte: A autora

10

Processo

Os serviços que o sistema operacional implementa através dos processos são: auditorias e segurança, serviços de rede, contabilização do uso de recursos e de erros, gerência de impressão, de processos em lote tipo *batch*, a temporização de processos, a comunicação entre eventos e, ainda, a interface de comandos (*shell*).

Os processos do S.O. que administram a comunicação entre os eventos e a sua sincronização ocorrem através do envio de sinais, que são bits que compõem o bloco de controle de processos (PCB).



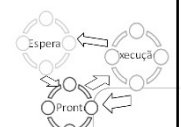
Fonte: A autora

11

Processo

Criação: 1. Início do sistema; 2. Execução de uma chamada de sistema de criação de processo por um processo em execução; 3. Uma requisição do usuário para criar um novo processo; 4. Início de uma tarefa em lote (*batch/job*).

Encerramento: 1. Há a saída normal ou voluntária do processo; 2. Há a saída por erro, que também é voluntária; 3. Ocorreu algum erro considerado fatal para a continuidade de execução do processo, neste caso involuntário; 4. Quando ocorre o cancelamento de um processo por uma solicitação de outro processo, que também é involuntário.



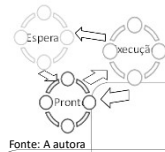
Fonte: A autora

12

Processo

Os processos são classificados em dois tipos:

- CPU-bound: ocupa mais recursos da unidade central de processamento (UCP), ou seja, passa mais tempo em execução e pronto. Facilmente encontrado em aplicações com maior quantidade de operações de cálculo.
- I/O-bound: este processo passa a maior parte do tempo em estado de espera. Encontrado em aplicações comerciais em que é necessário realizar muitas tarefas de leitura, gravação e processamento.



Fonte: A autora

13

Threads

Thread é um pequeno programa que trabalha como um subsistema, sendo uma forma de um processo se autodividir em duas ou mais tarefas. É o termo em inglês para Linha ou Encadeamento de Execução. Essas tarefas múltiplas podem ser executadas simultaneamente para rodar mais rápido do que um programa em um único bloco ou praticamente juntas, mas que são tão rápidas que parecem estar trabalhando em conjunto ao mesmo tempo.

14

Resolução da SP

- Identificar o *software* de gestão integrada e quais são as suas configurações básicas ou padrão;
- verificar quais são as especificações e se são compatíveis com as estações de trabalho;
- levantar dados de cache, *clock* e núcleos/*threads*;
- identificar quais são as chamadas de sistema para o gerenciamento de um dos sistemas operacionais identificados.

15

Configurações servidor e estações de trabalho

Recursos	Servidor	Estações de trabalho
Processador	Intel® Xeon® Processor E7-8830	Intel® Core™ M vPro™
Sistema Operacional	Linux CentOS 6.5	Windows XP SP3, Seven, 8 e 8.1
Cache	24 MB	4MB
Velocidade do Clock	2.13 GHz	900 MHz
Núcleos/Threads	8/16	2/4

Fonte: Elaborada pela autora

16

Tipos de chamadas de sistema em Linux

Chamada de sistema	Descrição
<code>pid = fork()</code>	Cria um processo filho idêntico ao pai
<code>pid = waitpid(pid, &status, opts)</code>	Espera o processo filho terminar
<code>s = execve(name, args, envp)</code>	Substitui a imagem da memória de um processo
<code>exit(status)</code>	Termina a execução de um processo e retorna o status
<code>s = sigaction(sig, &act, &oldact)</code>	Define a ação a ser tomada nos sinais
<code>s = sigreturn(&context)</code>	Retorna do um sinal
<code>s = sigprocmask(how, &set, &old)</code>	Examina ou modifica a máscara do sinal
<code>s = sigpending(&set)</code>	Obtém o conjunto de sinais bloqueados
<code>s = sigsuspend(&sigmask)</code>	Substitui a máscara de sinal e suspende o processo
<code>s = kill(pid, sig)</code>	Envia um sinal para um processo
<code>residual = alarm(seconds)</code>	Ajusta o relógio do alarme
<code>s = pause()</code>	Suspende o chamador até o próximo sinal

Fonte: Tanenbaum (2009)

17

Conceitos

Comunicação entre processos e problemas clássicos.

18

Comunicação

O S.O se comunica com o usuário de três formas:

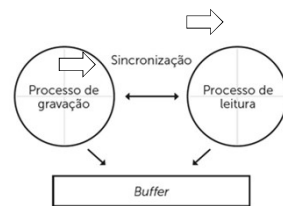
- através de procedimentos próprios do sistema,
- por meio da interação com os aplicativos
- através das linguagens de comando.

Tipo de comunicação:

- modo usuário - instruções não privilegiadas;
- modo kernel - o sistema operacional tem acesso irrestrito às instruções do processador.

19

Sincronização de leitura e gravação de processos



Fonte: Adaptado de Machado e Maia (2013, p. 94)

20

Comandos FORK e JOIN – Concorrência em programas

FORK - realiza uma chamada do processo que está no *buffer* para ser executado e, a partir da sua identificação, o associa ao seu subprocesso, ou seja, ao processo filho. Faz o acompanhamento de execução desse processo. **FORK** tem a finalidade de criar processos, o comando **JOIN** tem o objetivo de sincronizar os processos criados pelo **FORK**.

21

Comandos FORK e JOIN – Concorrência em programas

FORK – Inicia a execução de outro programa concorrentemente

JOIN – O programa chamador espera o outro programa terminar para continuar o processamento

PARBEGIN – Inicia lista de programas que serão executados paralela e aleatoriamente.

PAREND – Especifica o ponto de sincronização.

22

S.O multiprogramáveis – problemas

Os processos concorrentes pode apresentar alguns problemas quando se trata do compartilhamento de recursos.

- ✓ Compartilhamento de um arquivo em disco;
- ✓ Compartilhamento de uma variável na memória principal entre dois processos.

23

Exclusão Mútua

Esse mecanismo impede que dois ou mais processos sejam executados compartilhando o mesmo recurso simultaneamente.

Mecanismos de Sincronização que resolvem a

Exclusão Mútua:

- ✓ Sincronização condicional;
- ✓ Semáforos,
- ✓ Monitores;
- ✓ Troca de mensagens
- ✓ *Deadlock*

24

Resolução da SP

Localizar o endereço de solicitação de informações *on-line* do sistema de gestão da clínica médica e informar, a partir disso, os consultórios e clínicas mais próximos do usuário, de acordo com o seu plano de saúde, além de oferecer o serviço de discagem direta para a realização do agendamento.

25

Nós consideraremos cada ação que eles precisam tomar como sendo os processos que o sistema deverá executar, implementando uma solução que previna a ocorrência de *deadlocks* e *starvation*.
As solicitações de clínicas e consultórios próximos ao cliente (paciente ou usuário) representam os processos.
Para cada solicitação, será preciso alocar os recursos e com isso verificar a sua disponibilidade. A solicitação do usuário será executada apenas se houver recurso para alocar que esteja de fato com o *status* de disponível.

26

É estabelecida uma regra para priorizar o acesso à região crítica do programa para que aconteça a execução do processo.
Esse problema podemos resolver por semáforos.

27

Conceitos

Introdução ao escalonamento

28

Definições

De que forma o sistema operacional seleciona esses processos e a partir de quais critérios?
Escalação é a seleção do processo que será executado a partir do momento em que ele entra em estado de pronto e precisa efetivamente ser executado.

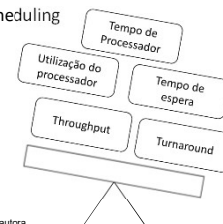


Fonte: A autora

29

Critérios de Escalonamento de Processos

Escalonador: Scheduling



Fonte: A autora

Processo possui:
Tempo de processador
Prioridade
Estado

30

Tipos de escalonamento

a) não preemptivos e preemptivos;	b) <i>first in first out</i> (FIFO);
c) <i>shortest job first</i> (SJF);	d) cooperativo;
e) circular;	f) por prioridades;
g) circular por prioridades;	h) por múltiplas filas;
i) por múltiplas filas com realimentação;	j) políticas de escalonamento em sistemas de tempo compartilhado e de tempo real.

Fonte: a autora

31

Resolução da SP

Sua tarefa é comprovar a eficiência do sistema operacional para realizar o gerenciamento dos processos do sistema de gestão integrado da aplicação da clínica médica de forma a otimizar, através das configurações de escalonamento, a execução dos processos solicitados pelo sistema.

32

Para resolver essa SP, vamos simular no SOSIM quatro processos (monitoramento de consultas, monitoramento de exames, controle de consultas e controle de exames).

- a) Escalonamento circular por prioridades;
- b) Crie quatro processos que tenham o mesmo nível de prioridade. Defina quais serão CPU-bound e quais serão I/O-bound;
- c) anote os tempos que os processos levam desde a criação até o encerramento, *turnaround*, bem como as mudanças de estados;

33

- d) Altere, também, a fatia de tempo que um processo pode levar;
 - e) Compare os tempos antes da alteração e depois, para poder dimensionar as respectivas mudanças de estados;
 - f) observe quais foram as variações em processamento, estados e tempo.
- Tome nota dos parâmetros utilizados e também dos resultados obtidos nesse processo. Correlacione com os processos do ERP.

34

Interação

Consegue analisar a eficiência do S.O?

35

Conceitos

Algoritmos de escalonamento

36

Escalonamento

Objetivo: Potencializar o uso de recursos da máquina:

- ✓ Maximizar o uso de recursos de forma a atender uma quantidade maior de processos e em menor tempo de execução.
- ✓ Minimizar os tempos de resposta
- ✓ Trabalhar com memória e processador de forma a evitar que aconteça a interrupção por tempo indefinido de um determinado processo.
- ✓ Prioridades na escolha de processos e alocação de recursos.

37

Outros Algoritmos de Escalonamento

Algoritmo de escalonamento	Descrição
Escalonamento por próxima taxa de escalonamento mais próxima.	Conhecido como HRRN (highest response ratio next). Visa corrigir um erro relacionado ao escalonamento SJR/SRT mencionado acima, em que há essencialmente a priorização de processos maiores a menores, fazendo com que esses tenham de esperar mais tempo do que o previsto.
Escalonamento por menor tempo de execução restante.	SRT (shortest remaining time), ao contrário do SJR, determina o processo que tem o menor tempo de execução, por estimativa e dá prioridade a esse.
Escalonamento por fração justa.	FJS (fair share scheduling) esse algoritmo permite que, em sistemas multiusuários, os processos relacionados a um usuário sejam agrupados aos de outros usuários para processamento. Também restringe cada grupo ao seu respectivo subconjunto de recursos do sistema.

Fonte: Adaptado de Dettel (2005, p. 218-228)

38

Outros Algoritmos de Escalonamento

Escalonamento por prazo	Esse tipo de algoritmo necessita da disponibilização extra de recursos necessários de processamento para que o sistema operacional possa distribuir sem prejudicar as rotinas comuns e desempenho. Dessa forma, o objetivo é que se cumpra o prazo predeterminado para o processamento do conjunto de dados.
Escalonamento de tempo real	Visa utilizar os recursos de processamento de forma potencializada, isto quer dizer que pode aplicar de acordo com a operação, um algoritmo diferente para cada um dos processos solicitados.
Escalonamento de thread Java	Esse tipo de escalonamento é realizado pelo sistema operacional e considera os níveis de suporte para threads, no entanto não distingue se é um processo multithread ou se deve trabalhar individualmente com os processos se este for de usuário e então necessita de uma biblioteca padrão para escalonar. Se for de núcleo, o sistema escolhe um por vez, de forma a considerar a alocação de recursos para cada um deles.

Fonte: Adaptado de Dettel (2005, p. 218-228)

39

Resolução da SP

Como avaliar um algoritmo que seja ideal para atender às necessidades tanto do sistema operacional quanto das aplicações que interagem no ambiente organizacional?

40

A avaliação de um algoritmo de escalonamento está diretamente relacionada à sua seleção. Ele deve ser adequado às necessidades de processamento e funcionalidades que devem ser aliadas ao sistema computacional. Desse modo, você precisa identificar quais são as necessidades de processamento do sistema. Para tal, podem ser elencados alguns pontos de atenção, como:

a) selecionar os critérios de escalonamento: orientado a processador, em lote, interativo, E/S;

41

Resolução da SP

Criar um quadro com as seguintes descrições:
Características técnicas , Como pode te ajudar e Valor da Licença

42

Interação

43

Banca: COPEVE-UFAL Órgão: MPE-AL Prova: COPEVE-UFAL - 2012 - MPE-AL - Analista do Ministério Público - Administração de Redes

São objetivos de um algoritmo de escalonamento para sistemas interativos:

- vazão e tempo de retorno.
- vazão e utilização da CPU.
- tempo de resposta e proporcionalidade.
- cumprimento dos prazos e previsibilidade.
- utilização da CPU e proporcionalidade.

44

Banca: FCC Órgão: DPE-SP Prova: FCC - 2015 - DPE-SP - Analista de Sistemas

Quando dois ou mais processos têm condições de "rodar", é o escalonador que decide, baseado em um algoritmo de escalonamento, qual será o próximo a receber tempo de CPU. Nesse contexto, quando há uma interrupção e suspensão temporária da execução de processos não bloqueados após um tempo máximo fixado tem-se o que categorizamos de escalonamento

- Preemptivo.
- First-Come, First Served.
- Hood-Robin.
- Quantum.
- Não-preemptivo.

45

Conceitos

Recapitulando

46

Definições

- ✓ Processos;
- ✓ Threads;
- ✓ Exclusão mútua;
- ✓ Escalonamento.

47



48