

Sistemas Operacionais

Aula 07 – Processos – SCC5854
Capítulo 5 (MACHADO; MAIA, 2014)

Prof. Dr. Jonathan Ramos
jonathan@unir.br

Departamento Acadêmico de Ciências de Computação – DACC

Núcleo de Tecnologia – NT

01/11/2022

Sumário

- 1 Estrutura do Processo
 - Contexto de Hardware
 - Contexto de Software
 - Espaço de Endereçamento
 - Bloco de Controle do Processo
- 2 Estados do Processo
- 3 Mudança de Estado do Processo
- 4 Criação e eliminação de Processos
- 5 Processos *CPU-bound* e *I/O-bound*
- 6 Processos *Foreground* e *Background*
- 7 Formas de Criação de Processos
- 8 Processos Independentes, Subprocessos e Threads
- 9 Processos do SO
- 10 Sinais

Introdução

Um programa ao ser executado deve estar sempre associado a um processo:

O conceito de processo é a base para a implementação de um sistema multi-programável

Ou seja, um processo é um programa em execução!

Introdução

Gerência de Processos:



Processos:

São executados concorrentemente;
É uma das principais funções de um SO.



Compartilhando o uso do processador:

Memória principal e dispositivos de E/S, etc



Possibilidade de execução simultânea

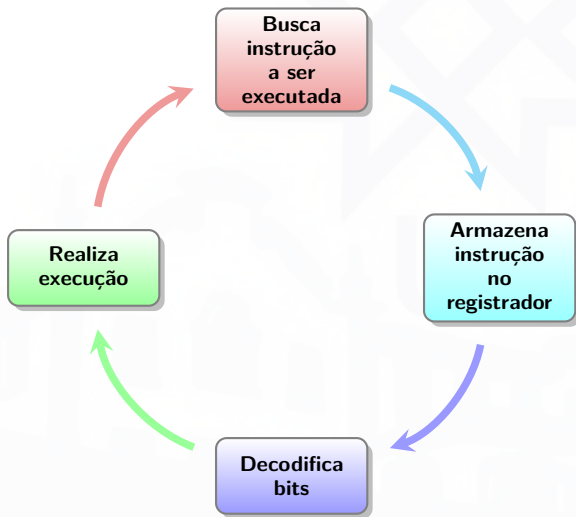
Dos processos nos diferentes processadores

Sumário

- 1 Estrutura do Processo
 - Contexto de Hardware
 - Contexto de Software
 - Espaço de Endereçamento
 - Bloco de Controle do Processo
- 2 Estados do Processo
- 3 Mudança de Estado do Processo
- 4 Criação e eliminação de Processos
- 5 Processos *CPU-bound* e *I/O-bound*
- 6 Processos *Foreground* e *Background*
- 7 Formas de Criação de Processos
- 8 Processos Independentes, Subprocessos e Threads
- 9 Processos do SO
- 10 Sinais

Estrutura do Processo

Revisão do “ciclo de busca e instrução”:



Em resumo:

A instrução deve:

- 1** ser buscada na memória principal;
- 2** executada pela CPU.

Estrutura do Processo

Na Visão do hardware:

- Instruções são executadas **sem saber qual programa está em processamento**.
- **É responsabilidade do SO** gerenciar concorrência de forma controlada e segura.
 - **Alternância na execução e uso da UCP.**



O conceito de Processo:

É essencial para sistemas multi-programáveis;
Implementem concorrência de diversos programas e diferentes usuários.



O Processo:

Pode ser entendido como um programa em execução;
Porém, é um pouco mais abrangente.

Estrutura do Processo

Para a concorrência de processos seja bem sucedida:

- **Todas as informações** de um processo deve ser mantida;
- Interrupções, ao retomar **deve continuar de onde parou**;
- Com isso, o **SO consegue gerenciar processos** de forma concorrente;
- **Base para qualquer sistema multi-programável.**



Processo de uma forma mais ampla:

É o conjunto necessário de informações para que o SO implemente a concorrência de programas

Concorrência de três programas (PROG₁, PROG₂, PROG₃) associados aos Processos X, Y e Z:

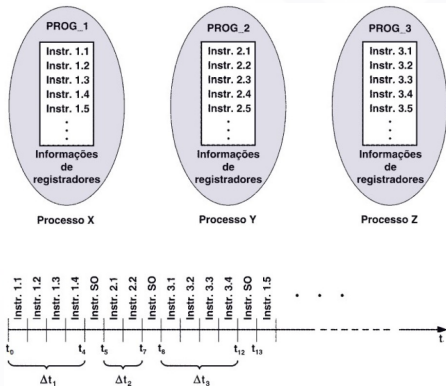


Figura: Concorrência de programas e processos.

- 1 Δt_1 : processador executa instruções do PROG₁: **em t_4 o SO interrompe temporariamente a execução do PROG₁** e salva o conteúdo dos registradores do processador, armazenando-os no **Processo X**.
- 2 Δt_2 : PROG₂ é iniciado e **em t_7 o SO interrompe PROG₂** e salva o conteúdo dos registradores no **Processo Y**.
- 3 Δt_2 : PROG₃ é iniciado e **em t_{12} é interrompido pelo SO**, salva seus registradores no **Processo Z** e retomar a execução de PROG₁.

Em t_{12} o conteúdo dos registradores do **Processo X** é carregado no processador:

PROG₁ continua sua execução **como se não tivesse sido interrompido**.

Estrutura do Processo

Mudança de contexto:

É a troca de um processo por outro no processador, comandada pelo SO:

- É dessa maneira que o SO implementa e gerencia um ambiente multi-programável.

Cada usuário tem seu programa associado a um processo:

Ao executar um programa, o usuário tem a impressão de possuir o processador e todos os demais recursos reservados exclusivamente para ele.

- Isto não é verdade, visto que todos os recursos estão sendo compartilhados, inclusive a UCP.
- Processador executa o programa de um usuário durante um intervalo de tempo e, conforme observado, no instante seguinte estará processando um outro programa

Estrutura do Processo



Estrutura do Processo

Um processo é formado por três partes:



Figura: Estrutura do processo.

Juntos mantêm todas as informações necessárias à execução de um programa:

Contexto de Hardware:

Armazena o **conteúdo dos registradores gerais da UCP**, além dos registradores de uso específico, como *program counter* (PC), *stack pointer* (SP) e registrador de status.

Contexto de Software:

São especificados **limites e características dos recursos que podem ser alocados pelo processo**, como o número máximo de arquivos abertos simultaneamente, prioridade de execução e tamanho do *buffer* para operações de E/S

Contexto de Endereçamento:

É a área de memória **pertencente ao processo onde instruções e dados do programa** são armazenados para execução

Contexto de Hardware – CH: O SO gerencia a Mudança de Contexto

Salvando o conteúdo dos registradores do processo:

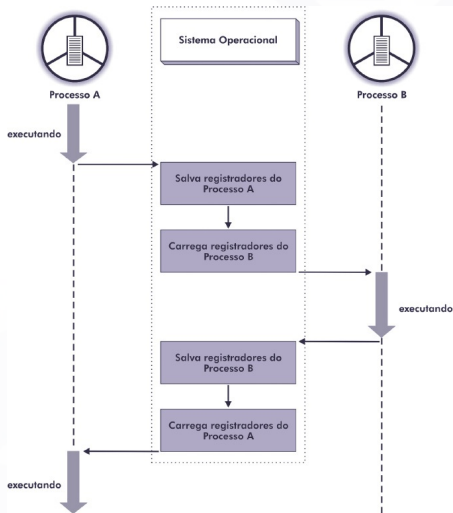


Figura: Mudança de contexto.

Execução/Pausa

- 1 Em um processo em execução, o seu CH está armazenado nos registradores do processador:
- 2 No momento em que o processo perde a utilização da UCP, o sistema salva as informações no CH do processo.

O CH é fundamental:

- Para a implementação dos sistemas multiprogramáveis;
- Os processos se alternam na utilização da UCP, podendo ser interrompidos e, posteriormente, restaurados

Em resumo: o CH é substituído entre um processo e outro.

Contexto de Software

Define o número máximo de arquivos abertos simultaneamente, prioridade de execução e tamanho do *buffer* para operações de E/S.



Identificação:

Cada processo recebe uma identificação única (PID - *process identification*)
Identificação única no sistema (UID — *user identification*),



Quotas:

Limites de cada recurso do sistema que um processo pode alocar.
Por exemplo, tamanho máximo de memória principal e secundária que o processo pode alocar.



Privilégios:

Ações que um processo pode fazer em relação a ele mesmo, aos demais processos e ao SO.
No sistema Unix existe a conta “root”, no MS Windows a conta “administrator”

Espaço de Endereçamento

É a área de memória pertencente ao processo onde instruções e dados do programa são armazenados para execução:



Cada processo:

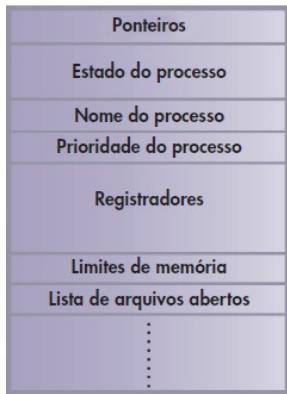
Possui seu próprio espaço de endereçamento, que deve ser devidamente protegido do acesso dos demais processos.

Figura: Características da estrutura de um processo.

Bloco de Controle do Processo

(Process Control Block — PCB)

O processo é implementado pelo SO através de uma estrutura de dados chamada bloco de controle do processo PCB.



Controle do processo:

O SO mantém todas as informações sobre o contexto de hardware, contexto de software e espaço de endereçamento de cada processo

PCBs:

São mantidos na memória principal em uma área exclusiva do SO.

Figura: Bloco de Controle do Processo (PCB).

Bloco de Controle do Processo

Visualização de processos no Linux:

#	ps	-l	-A									
F S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4 S	0	1	0	0	75	0	-	378	schedu ?		00:00:04	init
1 S	0	2	1	0	75	0	-	0	contex ?		00:00:00	keventd
1 S	0	3	1	0	94	19	-	0	ksofti ?		00:00:00	ksoftirqd/0
1 S	0	6	1	0	85	0	-	0	bdfus ?		00:00:00	bdfush
1 S	0	4	1	0	75	0	-	0	schedu ?		00:05:35	kswapd
1 S	0	5	1	0	75	0	-	0	schedu ?		00:03:45	kscand
1 S	0	7	1	0	75	0	-	0	schedu ?		00:00:00	kupdated
1 S	0	8	1	0	85	0	-	0	md_thr ?		00:00:00	mrecoveryd
1 S	0	21	1	0	75	0	-	0	end ?		00:05:40	kjournald
1 S	0	253	1	0	75	0	-	0	end ?		00:00:00	kjournald
1 S	0	254	1	0	75	0	-	0	end ?		00:00:00	kjournald
1 S	0	255	1	0	75	0	-	0	end ?		00:55:28	kjournald
1 S	0	579	1	0	75	0	-	399	schedu ?		00:02:00	syslogd
5 S	0	583	1	0	75	0	-	383	do_sys ?		00:00:00	klogd
5 S	32	600	1	0	75	0	-	414	schedu ?		00:00:00	portmap
5 S	29	619	1	0	85	0	-	416	schedu ?		00:00:00	rpc.statd
1 S	0	631	1	0	75	0	-	393	schedu ?		00:00:00	mdadm
5 S	0	702	1	0	75	0	-	917	schedu ?		00:00:30	sshd
5 S	0	716	1	0	75	0	-	539	schedu ?		00:00:00	xinetd
5 S	0	745	1	0	75	0	-	398	schedu ?		00:00:00	gpm
5 S	0	765	1	0	75	0	-	607	schedu ?		00:00:16	cron

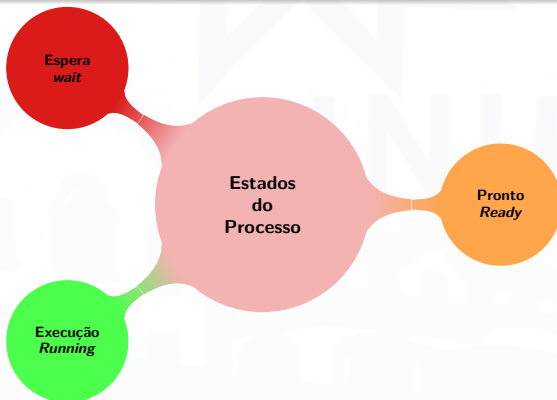
Toda a gerência dos processos é realizada por intermédio de chamadas a rotinas do SO que realizam operações como criação, alteração de características, visualização, eliminação, sincronização, suspensão de processos, dentre outras.

Estados do Processo

Um processo não deve alocar exclusivamente a UCP:

Deve existir um compartilhamento no uso do processador.

- Os processos passam por diferentes estados ao longo do seu processamento, em função de eventos gerados pelo sistema operacional ou pelo próprio processo.



Estados do Processo: Em execução – *RUNNING*

Em execução – *RUNNING*

Está sendo processado pela UCP:

- **Um único processador:** somente um processo pode estar sendo executado em um dado instante de tempo
 - Os processos se alternam na utilização do processador seguindo uma política estabelecida pelo SO
- **Vários processadores:** existe a possibilidade de mais de um processo ser executado ao mesmo tempo.
 - Também é possível um mesmo processo ser executado simultaneamente em mais de uma UCP (processamento paralelo).

Estados do Processo: Pronto – *READY*

Pronto – *READY*

Aguarda apenas para ser executado:

- **Escalonamento:** O SO é responsável por determinar a ordem e os critérios pelos quais os processos em estado de pronto devem fazer uso do processador
- **Vários processos no sistema no estado de pronto:** organizados em listas encadeadas
 - Os processos devem estar ordenados pela sua importância;
 - Permitir que processos mais prioritários sejam selecionados primeiramente para execução;

Estados do Processo: Espera – *WAIT*

Espera – *WAIT*

aguarda por algum evento externo ou por algum recurso para prosseguir seu processamento:

- **Exemplo:** processo está esperando o término de uma operação de entrada/saída ou a espera de uma determinada data e/ou hora para continuar sua execução;
- Em alguns sistemas operacionais, o estado de espera pode ser chamado de **bloqueado (*blocked*)**.
- O sistema organiza os vários processos no estado de espera também em listas encadeadas

Sumário

- 1 Estrutura do Processo
 - Contexto de Hardware
 - Contexto de Software
 - Espaço de Endereçamento
 - Bloco de Controle do Processo
- 2 Estados do Processo
- 3 Mudança de Estado do Processo
- 4 Criação e eliminação de Processos
- 5 Processos *CPU-bound* e *I/O-bound*
- 6 Processos *Foreground* e *Background*
- 7 Formas de Criação de Processos
- 8 Processos Independentes, Subprocessos e Threads
- 9 Processos do SO
- 10 Sinais

Mudança de Estado do Processo

Um processo muda de estado durante seu processamento em função de:

- **Eventos voluntários:** eventos originados por ele próprio;
- **Eventos involuntários:** ou pelo SO.

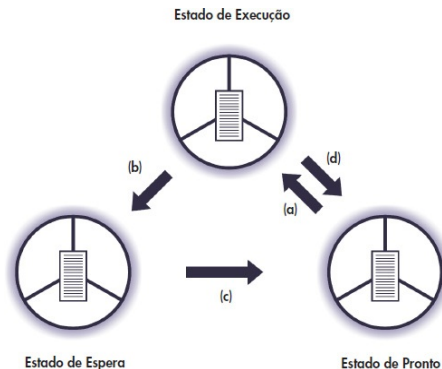


Figura: Mudanças de estado do processo (1).

Mudança de Estado do Processo

Pronto => Execução:

Cada sistema operacional tem seus próprios critérios e algoritmos para a escolha da ordem em que os processos serão executados (política de escalonamento):

- Esses critérios e seus algoritmos serão analisados no Capítulo 8.

Execução => Espera:

Por eventos gerados pelo próprio processo:

- Uma operação de E/S;
- Por eventos externos.

Mudança de Estado do Processo

Espera => Pronto:

Quando a operação solicitada é atendida ou o recurso esperado é concedido:

- Um processo no estado de espera sempre terá de passar pelo estado de pronto antes de poder ser novamente selecionado para execução;
- Não existe a mudança do estado de espera para o estado de execução diretamente.

Execução => Pronto:

Por eventos gerados pelo sistema, como o término da fatia de tempo que o processo possui para sua execução:

- O processo volta para a fila de pronto, onde aguarda por uma nova oportunidade para continuar seu processamento.

Sumário

- 1 Estrutura do Processo
 - Contexto de Hardware
 - Contexto de Software
 - Espaço de Endereçamento
 - Bloco de Controle do Processo
- 2 Estados do Processo
- 3 Mudança de Estado do Processo
- 4 Criação e eliminação de Processos**
- 5 Processos *CPU-bound* e *I/O-bound*
- 6 Processos *Foreground* e *Background*
- 7 Formas de Criação de Processos
- 8 Processos Independentes, Subprocessos e Threads
- 9 Processos do SO
- 10 Sinais

Criação e eliminação de Processos

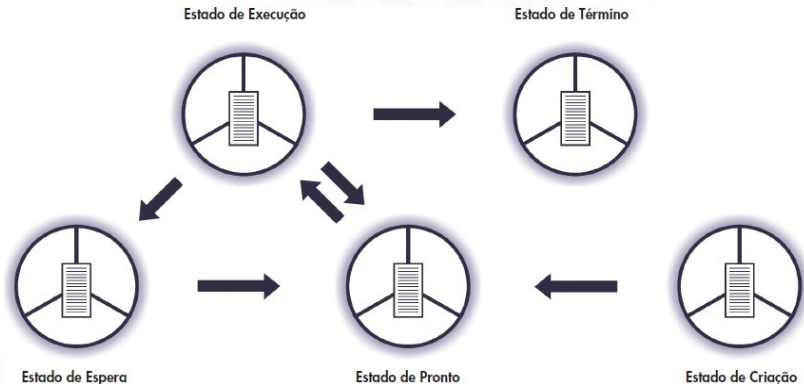


Figura: Mudanças de estado do processo (3).

Criação (new)

O SO já cria um PCB, porém não adicionar aos processos prontos:

- Alguns SO podem manter processos no estado de criação devido a recursos disponíveis.

Criação e eliminação de Processos

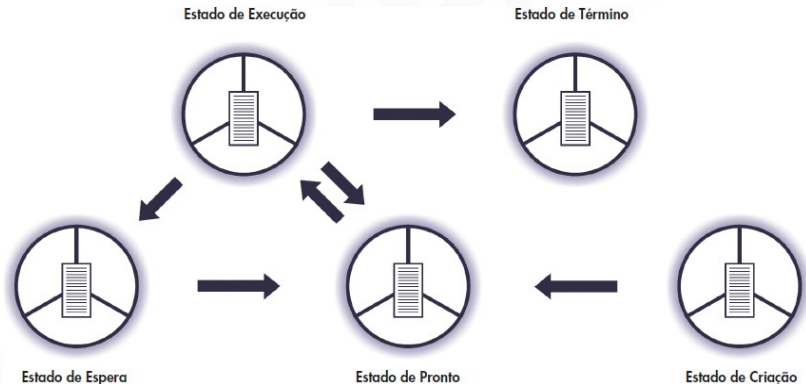


Figura: Mudanças de estado do processo (3).

Terminado (*exit*)

Não poderá ter mais nenhum programa executado no seu contexto:

- O SO ainda mantém suas informações de controle presentes em memória;

Criação e eliminação de Processos

O término de processo pode ocorrer por motivos como:

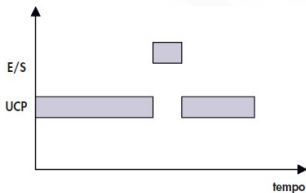
- 1 **Término normal de execução;**
- 2 **Eliminação por um outro processo;**
- 3 **Eliminação forçada por ausência de recursos disponíveis no sistema.**

Sumário

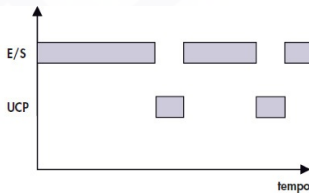
- 1 Estrutura do Processo
 - Contexto de Hardware
 - Contexto de Software
 - Espaço de Endereçamento
 - Bloco de Controle do Processo
- 2 Estados do Processo
- 3 Mudança de Estado do Processo
- 4 Criação e eliminação de Processos
- 5 Processos *CPU-bound* e *I/O-bound***
- 6 Processos *Foreground* e *Background*
- 7 Formas de Criação de Processos
- 8 Processos Independentes, Subprocessos e Threads
- 9 Processos do SO
- 10 Sinais

Processos *CPU-bound* e *I/O-bound*

Processos podem ser classificados de acordo com a utilização do processador e E/S:



(a) CPU-bound



(b) I/O-bound

(a) *CPU-bound* (ligado à UCP):

Passa a maior parte do tempo em execução, utilizando o processador, ou pronto:

- Realiza **poucas operações de leitura e gravação**.
- É encontrado em **aplicações científicas** que efetuam **muitos cálculos**.

(b) *I/O-bound* (ligado à E/S):

Passa a maior parte do tempo em espera,
pois realiza um elevado número de
operações de E/S:

- É encontrado em **aplicações comerciais**, que se baseiam em **leitura, processamento e gravação**;
- Processos interativos, **exemplos ?**;

Sumário

- 1 Estrutura do Processo
 - Contexto de Hardware
 - Contexto de Software
 - Espaço de Endereçamento
 - Bloco de Controle do Processo
- 2 Estados do Processo
- 3 Mudança de Estado do Processo
- 4 Criação e eliminação de Processos
- 5 Processos *CPU-bound* e *I/O-bound*
- 6 Processos *Foreground* e *Background***
- 7 Formas de Criação de Processos
- 8 Processos Independentes, Subprocessos e Threads
- 9 Processos do SO
- 10 Sinais

Processos *Foreground* e *Background*

(a) Processo *Foreground*



(b) Processo *Background*



Figura: Processos *foreground* e *background*.

Processos *Foreground* e *Background*

Processo *foreground*:

É aquele que permite a comunicação direta do usuário com o processo durante o seu processamento:

- Tanto o canal de entrada quanto o de saída estão associados a um terminal com teclado, mouse e monitor;
- Permite a interação com o usuário.

Processo *background*:

É aquele onde não existe a comunicação com o usuário durante o seu processamento:

- Os canais de E/S não estão associados a nenhum dispositivo de E/S interativo, mas em geral a arquivos de E/S.
- É possível associar o canal de saída de um processo ao canal de entrada de um outro processo;

Processos *Foreground* e *Background*

É possível associar o canal de saída de um processo ao canal de entrada de outro:

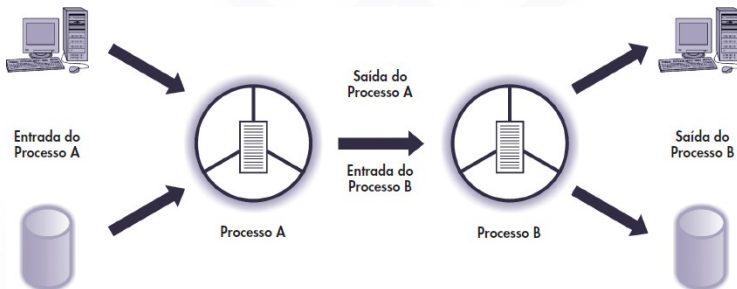


Figura: Pipe.

Neste caso:

Dizemos que existe um *pipe* ligando os dois processos:

- Se um **Processo A** **gera uma listagem** e o **Processo B** **tem como função ordená-la**, basta **associar o canal de saída do Processo A ao canal de entrada do Processo B**.

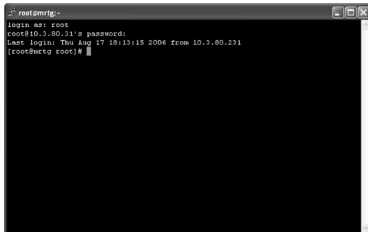
Formas de Criação de Processos

A seguir são apresentadas as três principais formas de criação de processos:

1 – Logon Interativo:

o usuário, no terminal, fornece ao sistema um *username* ou logon e um *password*:

- 1** O SO autentica estas informações no **arquivo de usuários**;



- 2 Em caso positivo, um processo *foreground* é criado;
 - Possibilita ao usuário interagir com o SO em uma linguagem de comandos.
- 3 O arquivo de usuários é um arquivo do SO onde são armazenados todos os usuários autorizados a ter acesso ao sistema.
- 4 Grande parte das informações do **contexto de software** de um processo provém do arquivo de usuários

Formas de Criação de Processos

2 – Via Linguagem de Comandos:

Um usuário **pode criar novos processos por intermédio de comandos da linguagem de comandos:**

- O processo criado pode ser *foreground* ou *background*, dependendo do comando de criação utilizado.

3 – Via Rotina do Sistema Operacional:

Pode ser criado a partir de qualquer programa executável com o uso de rotinas do SO:

- Possibilita a execução de outros programas concorrentemente ao programa chamador;
- A rotina de criação de processos depende do sistema operacional e possui diversos parâmetros:
 - nome do processo a ser criado;
 - nome do programa executável que será executado dentro do contexto do processo;
 - prioridade de execução;
 - estado do processo;
 - Se o processo é do tipo *foreground* ou *background* etc

Sumário

- 1 Estrutura do Processo
 - Contexto de Hardware
 - Contexto de Software
 - Espaço de Endereçamento
 - Bloco de Controle do Processo
- 2 Estados do Processo
- 3 Mudança de Estado do Processo
- 4 Criação e eliminação de Processos
- 5 Processos *CPU-bound* e *I/O-bound*
- 6 Processos *Foreground* e *Background*
- 7 Formas de Criação de Processos
- 8 Processos Independentes, Subprocessos e Threads**
- 9 Processos do SO
- 10 Sinais

Processos Independentes, Subprocessos e Threads

São maneiras diferentes de implementar a concorrência em uma aplicação:

Forma cooperativa:

Busca-se subdividir o código em partes.

Exemplo:

Um banco de dados com produtos de uma grande loja, onde vendedores fazem consultas frequentes:

- A concorrência na aplicação proporciona um tempo de espera menor entre as consultas, melhorando o desempenho da aplicação e beneficiando os usuários

Processos independentes:

É a maneira mais simples de implementar a concorrência em sistemas multiprogramáveis:

- Não existe vínculo do processo criado com o seu criador;
- exige a alocação de um PCB

Processos Independentes, Subprocessos e Threads

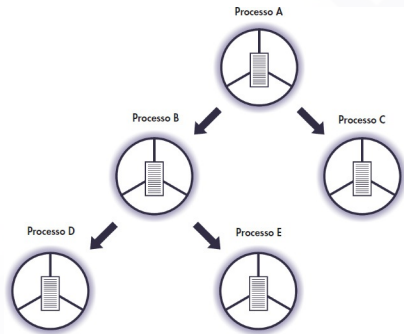


Figura: Estrutura de processos e subprocessos.

Subprocessos:

Criados dentro de uma estrutura hierárquica.

- O processo criador é denominado processo-pai, enquanto o novo processo é chamado de subprocesso ou processo-filho;
- Uma característica desta implementação é a dependência existente entre o processo criador e o subprocesso
- Caso um processo-pai deixe de existir, os subprocessos subordinados são automaticamente eliminados
- subprocessos possuem seu próprio PCB

Processos Independentes, Subprocessos e Threads

Desvantagens:

Demanda consumo de diversos recursos do sistema:

- Sempre que um novo processo é criado, o sistema deve alocar recursos (contexto de hardware, contexto de software e espaço de endereçamento), consumindo tempo de UCP neste trabalho
- No momento do término dos processos, o sistema operacional também dispensa tempo para desalocar recursos previamente alocados
- Outro problema é a comunicação e a sincronização entre processos consideradas pouco eficientes

Processos do Sistema Operacional

São Processos criados e gerenciados pelo próprio SO.

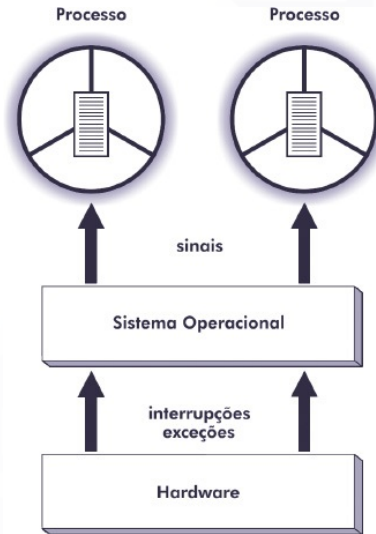
Serviços que o sistema operacional pode implementar:

- 1 Auditoria e segurança;
- 2 Serviços de rede;
- 3 Contabilização do uso de recursos;
- 4 Contabilização de erros;
- 5 Gerência de impressão;
- 6 Gerência de *jobs batch*;
- 7 Temporização;
- 8 Comunicação de eventos;
- 9 Interface de comandos (*shell*).

Sumário

- 1 Estrutura do Processo
 - Contexto de Hardware
 - Contexto de Software
 - Espaço de Endereçamento
 - Bloco de Controle do Processo
- 2 Estados do Processo
- 3 Mudança de Estado do Processo
- 4 Criação e eliminação de Processos
- 5 Processos *CPU-bound* e *I/O-bound*
- 6 Processos *Foreground* e *Background*
- 7 Formas de Criação de Processos
- 8 Processos Independentes, Subprocessos e Threads
- 9 Processos do SO
- 10 Sinais

Sinais



Mecanismo de sinais:

Assemelha-se ao tratamento de interrupções e exceções vistos no Capítulo 3 — Concorrência:

- Porém possui propósitos diferentes;
- O sinal está para o processo assim como as interrupções e exceções estão para o SO.

Figura: Sinais, interrupções e exceções.

Exercícios I

- 1 Defina o conceito de processo com suas palavras.
- 2 Por que o conceito de processo é tão importante no projeto de sistemas multi-programáveis?
- 3 É possível que um programa execute no contexto de um processo e não execute no contexto de um outro? Por quê?
- 4 Quais partes compõem um processo?
- 5 O que é o contexto de hardware de um processo e como é a implementação da troca de contexto?
- 6 Qual a função do contexto de software? Exemplifique cada grupo de informação.
- 7 O que é o espaço de endereçamento de um processo?
- 8 Como o sistema operacional implementa o conceito de processo? Qual a estrutura de dados indicada para organizar os diversos processos na memória principal?
- 9 Defina os cinco estados possíveis de um processo.
- 10 Dê um exemplo que apresente todas as mudanças de estado de um processo, juntamente com o evento associado a cada mudança.

Exercícios II

- 11 Diferencie processos multithreads, subprocessos e processos independentes.
- 12 Explique a diferença entre processos foreground e background.
- 13 Qual a relação entre processo e arquitetura microkernel?
- 14 Dê exemplos de aplicações CPU-bound e I/O-bound.
- 15 Justifique com um exemplo a frase "o sinal está para o processo assim como as interrupções e exceções estão para o sistema operacional".
- 16 Explique como a eliminação de um processo utiliza o mecanismo de sinais.

Laboratório com o sistema SOsim

Atividade 1: Criação de Processos

1 Práticas de simulação:

- Execute o simulador SOsim e identifique as quatro janelas que são abertas na inicialização.
- Crie um processo: janela Gerência de Processos / Criar – janela Criação de Processos / Criar.

2 Análise prática:

- Na janela Gerência de Processos, observe algumas informações sobre o contexto de software do processo como PID, prioridade, estado do processo e tempo de processador.
- Na janela Gerência de Processador, observe o processo transicionado entre estados.
- Na janela Gerência de Processador, movimente a barra de Clock de UCP e observe as variações ocorridas.

3 Questão teórica para responder com a ajuda do simulador:

- Com base na observação do comportamento do processo criado, identifique se o processo é I/O-bound ou CPU-bound. Justifique a resposta.

Laboratório com o sistema SOsim

Atividade 2: Tipos de Processos

1 Práticas de simulação

- Reinicialize o simulador.
- Crie um processo do tipo CPU-bound: janela Gerência de Processos / Criar – janela Criação de Processos / Criar (tipo de processo deve ser CPU-bound).
- Crie outro processo do tipo I/O-bound: janela Gerência de Processos / Criar – janela Criação de Processos / Criar (tipo de processo deve ser I/O-bound).

2 Análise prática:

- Na janela Gerência de Processos, observe as mudanças de estado dos dois processos.
- Na janela Gerência de Processador, observe o comportamento dos processos e as mudanças de contexto em função do tipo I/O-bound e CPU-bound.
- Na janela Gerência de Processos, compare a taxa de crescimento do tempo de processador dos dois processos.

3 Questão teórica para responder com a ajuda do simulador

- Analise os efeitos gerados no caso de redução do tempo gasto na operação de E/S pelo processo I/O-bound.

Laboratório com o sistema SOsim

Atividade 3: PCB

1 Práticas de simulação:

- Reinicialize o simulador.
- Crie dois novos processos: janela Gerência de Processos / Criar – janela Criação de Processos / Criar.

2 Análise prática:

- Na janela Gerência de Processos / PCB, observe as informações sobre o contexto de software e hardware dos processos criados.

3 Questão teórica para responder com a ajuda do simulador:

- Identifique quais informações do PCB são estáticas ou dinâmicas e quais fazem parte do contexto de software e do contexto de hardware.

Laboratório com o sistema SOsim

Atividade 4: Estatísticas

1 Práticas de simulação:

- Reinicialize o simulador.
- Ative a janela de Estatísticas em Console SOsim / Janelas / Estatísticas.
- Crie dois novos processos: janela Gerência de Processos / Criar – janela Criação de Processos / Criar.

2 Análise prática:

- Na janela Estatísticas, observe as informações: número de processos, estados dos processos e processos escalonados.

3 Questão teórica para responder com a ajuda do simulador:

- Observe que em alguns momentos existem processos no estado de pronto, porém nenhum em estado de execução. Explique a razão dessa situação.

Laboratório com o sistema SOsim

Atividade 5: Log de Execução dos Processos

1 Práticas de simulação:

- Reinicialize o simulador.
- Ative a janela de Log em Console SOsim / Janelas / Log.
- Crie dois novos processos do tipo CPU-bound: janela Gerência de Processos / Cria — janela Criação de Processos / Criar (tipo de processo deve ser CPU-bound).

2 Análise prática:

- Na janela Log, observe as informações sobre as mudanças de estado dos processos levando em conta o tempo que cada processo permanece nos estados de Execução e Pronto.
- Configure o simulador com um valor de fatia de tempo diferente.
- Reinicie o simulador SOsim para que a nova parametrização passe a ser válida.
- Observe as diferenças na janela Log.

3 Questão teórica para responder usando o simulador:

- Analise comparativamente a concorrência de dois processos CPU-bound executando em dois sistemas operacionais que se diferenciam apenas pelo valor da fatia de tempo.

Laboratório com o sistema SOsim

Atividade 6: Suspensão e Eliminação de Processos

1 Práticas de simulação:

- Reinicialize o simulador. Depois crie dois novos processos: janela Gerência de Processos / Cria – janela Criação de Processos / Criar.

2 Análise prática:

- Na janela Gerência de Processos, observe as informações sobre o contexto de software dos processos criados.
- Na janela Gerência de Processador, observe a concorrência no uso do processador pelos dois processos.
- Compare percentualmente os tempos de uso do processador entre os dois processos.
- Suspenda % um dos processos na janela Gerência de Processos/Suspend.
- Observe os estados dos processos, a concorrência no uso do processador e novamente compare percentualmente os tempos de uso do processador entre os dois processos.
- Libere o processo do estado de espera (suspenso) na janela Gerência de Processos/Prosseguir.
- Elimine um dos processos na janela Gerência de Processos/Finalizar.

3 Questão teórica para responder com a ajuda do simulador:

- Ao se eliminar um processo em estado de suspenso, o processo não é eliminado imediatamente. Reproduza essa situação no simulador e explique a razão da situação.

Referências I



MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistemas Operacionais**. 5a. ed. [S.l.]: Ed. LTC, 2014.

FIM!

`jonathan@unir.br`