

Sistemas Operacionais

Processos

Edeyson Andrade Gomes

edeyson@uol.com.br

Roteiro da Aula

- ▶ Conceitos
- ▶ Estados do Processo
- ▶ Controle de Processo
- ▶ Escalonamento
- ▶ Criação e Término de Processos

Definição

▶ Processo

- ▶ Um programa em execução
- ▶ Não é o mesmo que Programa (entidade estática)
 - ▶ Programa é o código executável
 - ▶ Processo é o código executando
- ▶ Entidade dinâmica
 - ▶ Ex.:A execução de prog.exe 3 vezes gera 3 processos distintos do mesmo programa

Definição

- ▶ Programa em execução e seu ambiente
 - ▶ Todo processo precisa ter:
 - ▶ Seção de texto
 - Código executável
 - ▶ Seção de dados
 - Variáveis, estruturas
 - ▶ Pilha do processo
 - Parâmetros, etc
 - ▶ Registradores, incluindo PC, SP, BP, etc

Conceitos

- ▶ Para que processos executem em ambiente multiprogramado existe a gerência de:
 - ▶ Compartilhamento da CPU, de memória, E/S, etc
- ▶ **Nomenclatura**
 - ▶ Sistemas Batch
 - ▶ Job
 - ▶ Sistemas de Tempo Compartilhado
 - ▶ Tarefa (on-line)

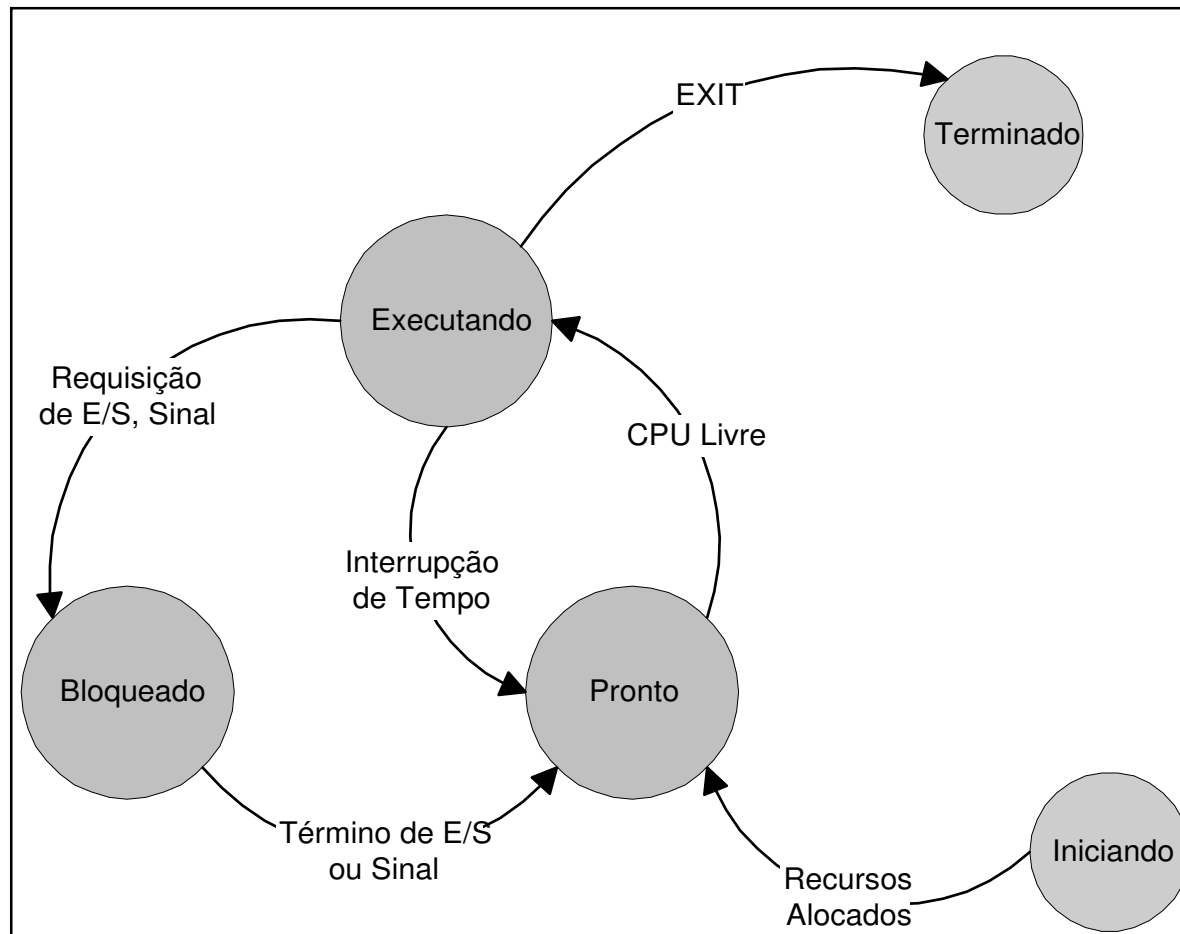
Estados do Processo

- ▶ Mudanças de estado durante a execução:
 - ▶ Iniciando
 - ▶ O processo está sendo criado
 - SO Aloca Memória, Contexto (BCP)
 - ▶ Ex.: O usuário *clica* num arquivo executável no Explorer ou executa via CMD
 - ▶ Executando
 - ▶ Instruções do processo estão sendo executadas
 - ▶ Bloqueado
 - ▶ O processo está esperando algum evento externo

Estados do Processo

- ▶ Mudanças de estado durante a execução:
 - ▶ Pronto
 - ▶ O processo está aguardando chance de ser executado no processador
 - ▶ Terminando
 - ▶ O processo está finalizando sua execução

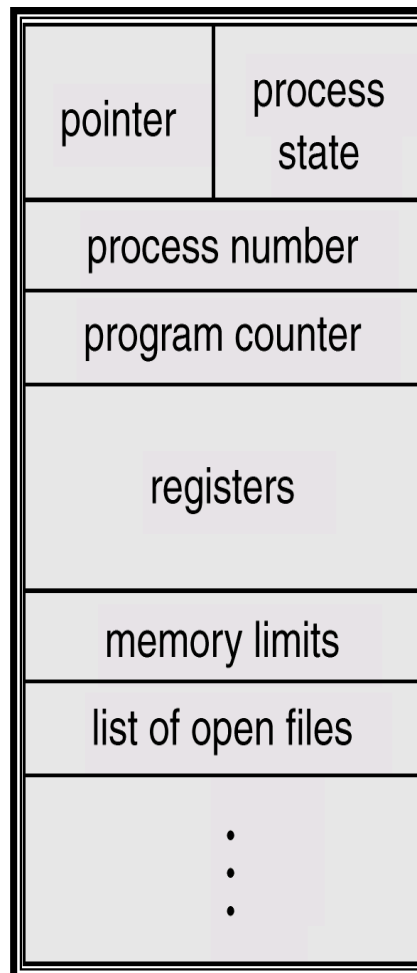
Estados do Processo



Controle de Processo

- ▶ **Bloco de Controle de Processo (BCP)**
 - ▶ Área de Memória alocada pelo S.O. para gerenciar processos
 - ▶ Mantém informações sobre o processo
 - ▶ Estado do processo
 - ▶ Registradores da CPU, incluindo o PC
 - ▶ Informações para escalonamento
 - ▶ Informações para gerenciamento de memória
 - ▶ Informações de contabilização
 - ▶ Informações sobre operações de E/S
 - ▶ Ponteiros para Arquivos, Socket, etc

Controle de Processo



Troca de Contexto

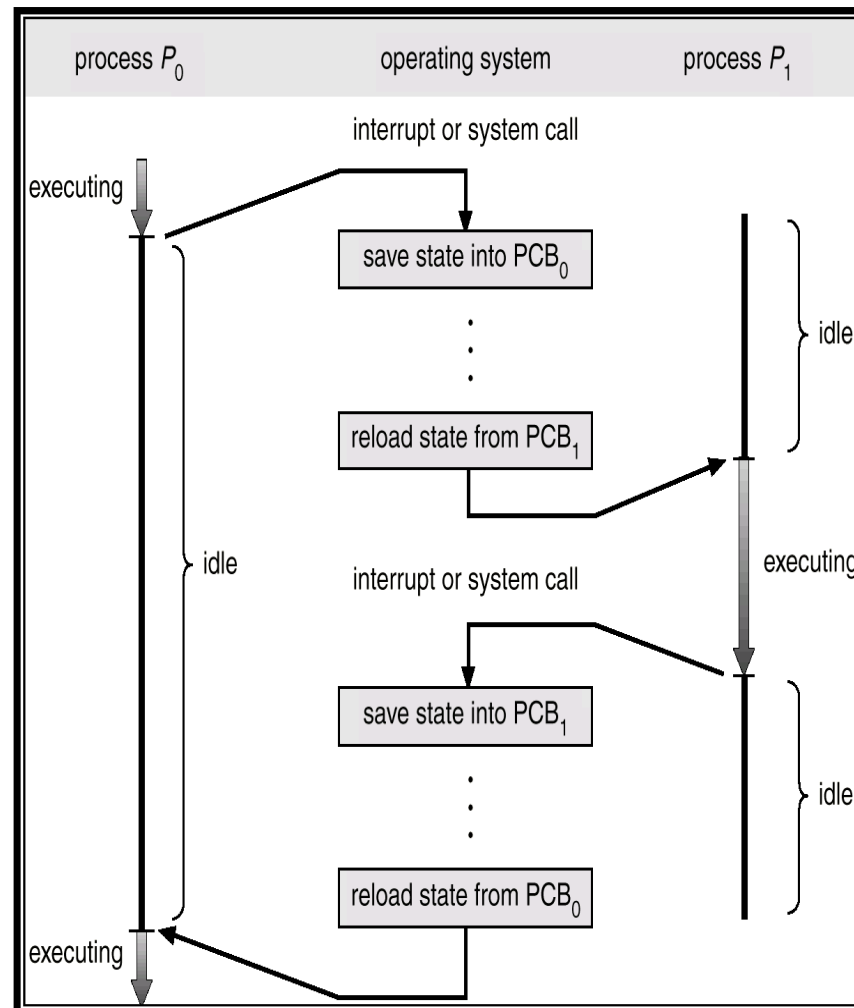
▶ Preempção

- ▶ Ação de retirar um processo da CPU a qualquer instante e restaurá-lo como se nada tivesse ocorrido
 - ▶ Suportado por Interrupções (como pelo CLOCK) e pelo BCP
 - Ex.: Preempção por tempo de Quantum
 - Preempção por I/O, Prioridade, etc.

▶ Tempo da troca é considerado overhead

▶ Tempo depende de suporte de hardware e complexidade do SO

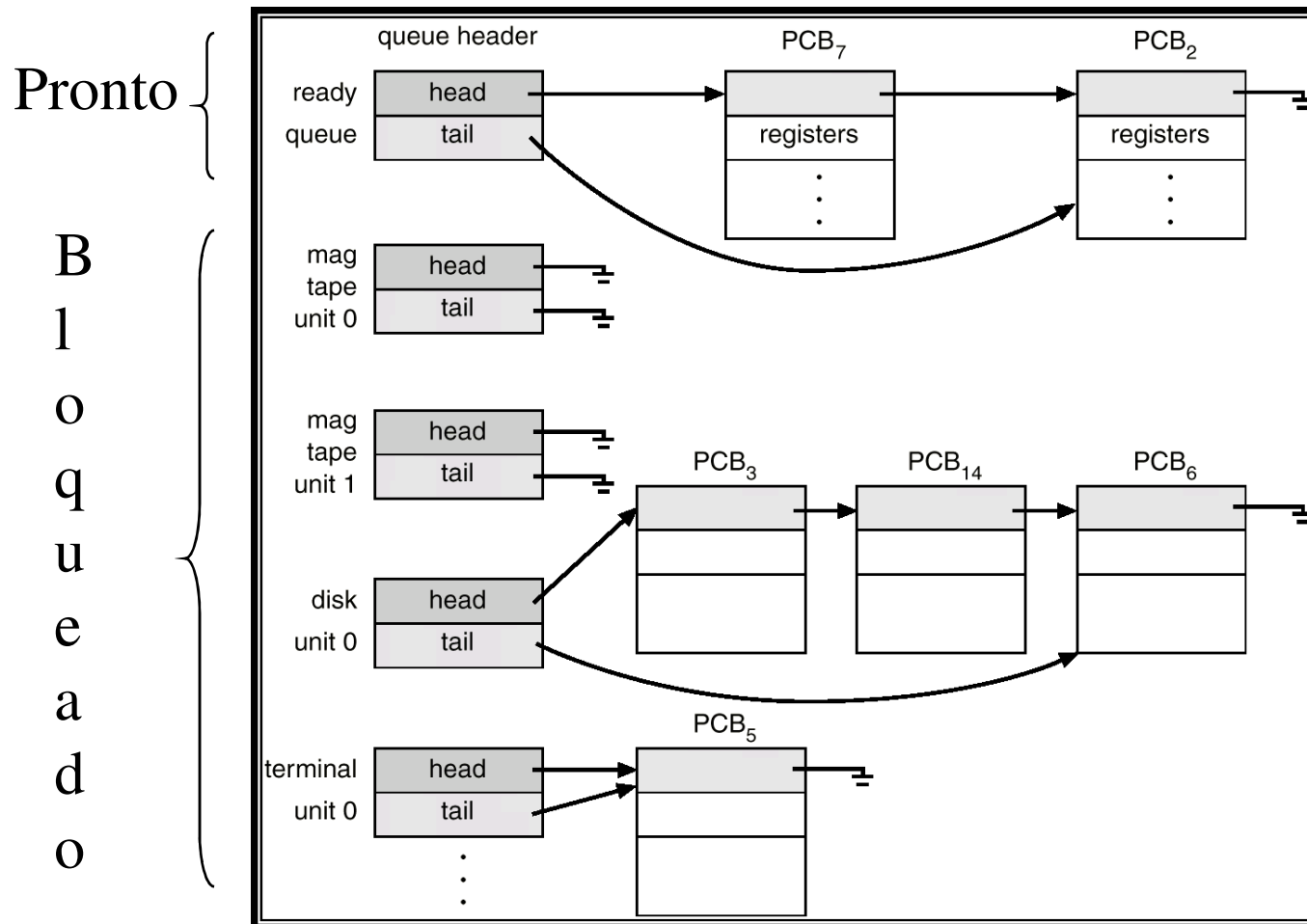
Troca de Contexto



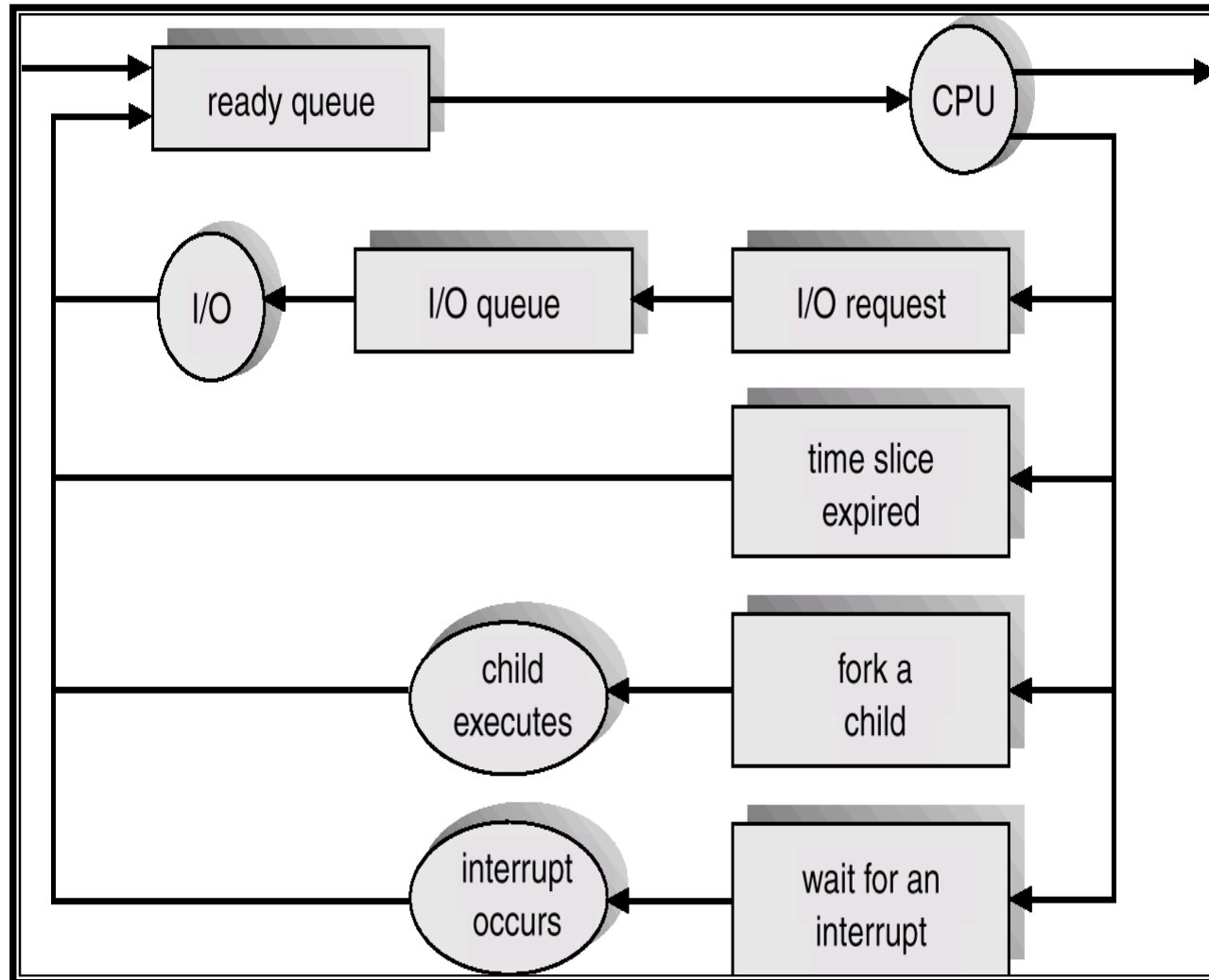
Escalonamento de Processos

- ▶ **Manutenção de filas para controle dos processos**
 - ▶ Fila de Processos – todos processos no sistema
 - ▶ Fila de Pronto – processos prontos
 - ▶ Filas de Bloqueado - processos aguardando E/S de dispositivo (interrupção) ou sinal
 - ▶ Múltiplas Filas – uma por dispositivo
 - Fila de Disco, Fila de CD, Fila de Teclado, de Rede
 - ▶ BCP efetua o encadeamento nas filas
 - ▶ Processos passam por várias filas

Escalonamento de Processos



Filas do Escalonamento



Escalonadores

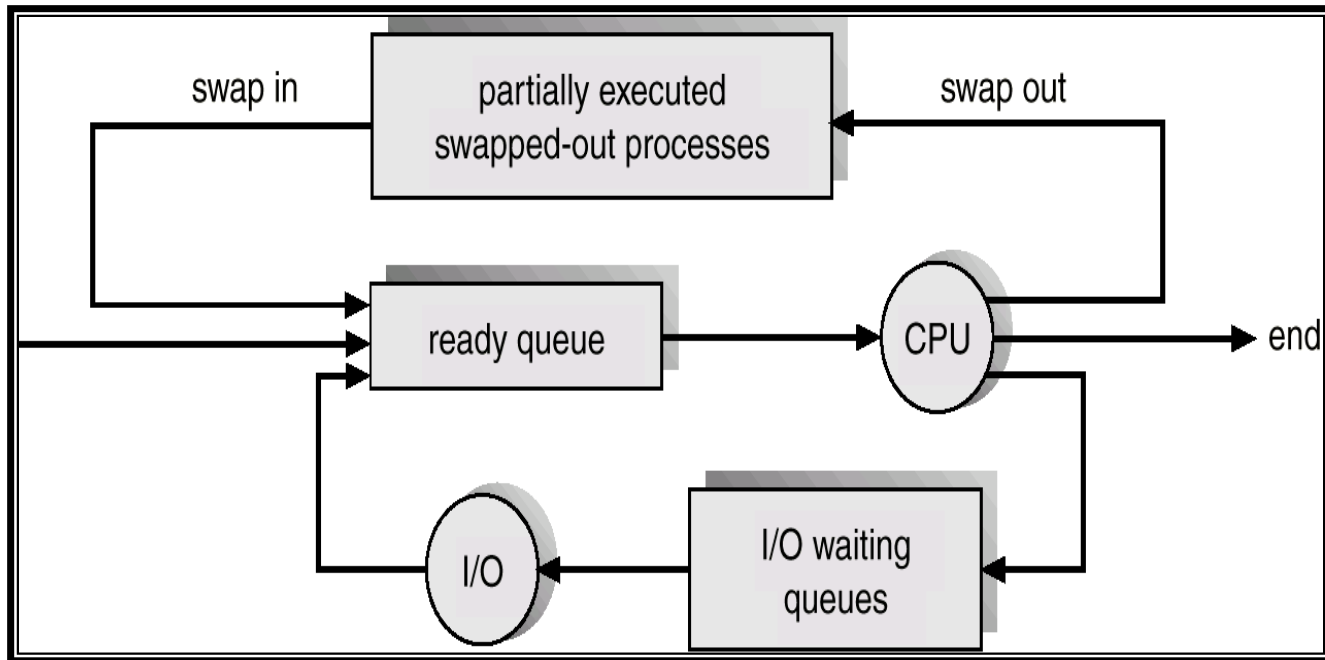
- ▶ Algoritmos (Políticas) de seleção
- ▶ Escalonador de Longo Prazo
 - ▶ Sistemas Batch, normalmente inexistente em sistemas de Tempo Compartilhado
 - ▶ JOBs submetidos em SPOOL
 - ▶ Seleciona JOB para ir para memória
 - ▶ Pronto

Escalonadores

- ▶ Escalonador de Curto Prazo (de CPU)
 - ▶ Seleciona processo pronto para uso de CPU

- ▶ Escalonador de Médio Prazo
 - ▶ Sistemas sem Job Scheduler, todo novo processo vai para memória
 - ▶ Problemas de desempenho
 - ▶ Remoção temporária de processos da memória

Escalonadores



Escalonadores

▶ Escalonadores de CPU (Curto Prazo)

- ▶ Executados freqüentemente (ms)
- ▶ Precisam ser rápidos
 - ▶ Diminuir overhead do S.O.

▶ Escalonadores de JOB (Longo Prazo)

- ▶ Executam com menos freqüência (seg., min.)
- ▶ Controlam o grau de multiprogramação
- ▶ Seleção deve ser cuidadosa
 - ▶ Efeito no desempenho do sistema

Escalonadores

- ▶ **Classificação de processos:**
 - ▶ **I/O-BOUND** – Intensivamente consumidor de E/S
 - ▶ Passa mais tempo fazendo operações de E/S do que utilizando a CPU
 - Processos Comercial
 - ▶ **CPU-BOUND** – Intensivamente consumidor de CPU
 - ▶ Passa mais tempo efetuando cálculos do que E/S
 - Processo Científico/Matemático
- ▶ **Híbrido**

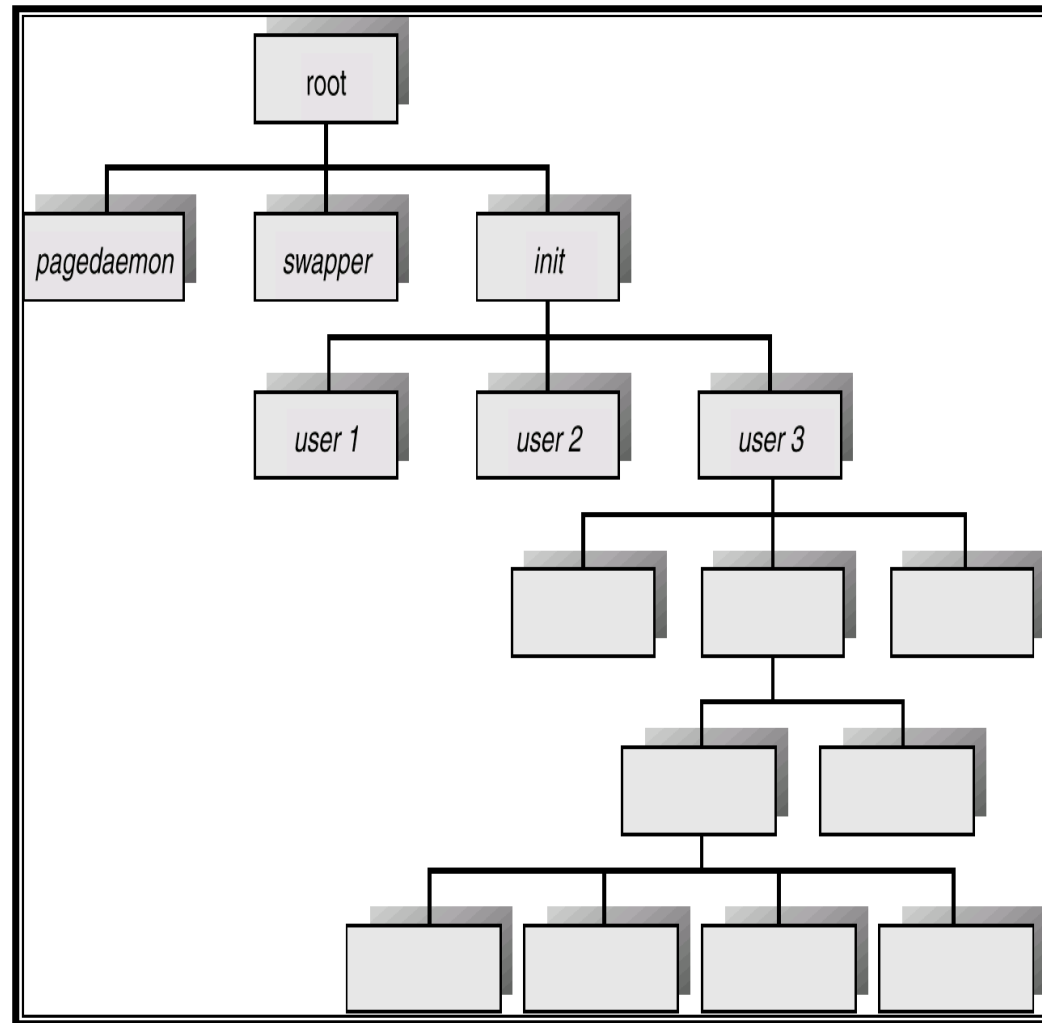
Escalonadores

- ▶ Combinação adequada de processos CPU e I/O BOUND melhora o desempenho e flexibilidade do sistema
 - ▶ Processo interativo (Ex.: Internet Explorer)
 - ▶ Tipicamente I/O Bound

Criação de Processos

- ▶ **Processo Pai, Filho e Árvore de Processos**
 - ▶ Processo Pai cria outro processo que se torna seu Filho
 - ▶ Identificação única dos processos
 - ▶ **PID – Process ID**

Criação de Processos



Criação de Processos

▶ Compartilhamento de Recursos

- ▶ Pai e Filho compartilham todos recursos
- ▶ Filho compartilha subconjunto dos recursos do Pai
- ▶ Pai e Filho não compartilham qualquer recurso

▶ Execução

- ▶ Pai e Filho executam concorrentemente
- ▶ Pai espera até que Filho termine

Criação de Processos

- ▶ **Espaço de Endereçamento**
 - ▶ Filho é duplicata do Pai
 - ▶ Unix, Windows
 - ▶ Filho tem um programa carregado
 - ▶ VMS, Windows

Criação de Processos

▶ Exemplo UNIX

- ▶ ***fork***: *system call* para criação de novo processo
 - ▶ Retorno “0” para processo filho
- ▶ ***exec***: *system call* para carregar novo programa no espaço de memória do processo
- ▶ ***wait***: *system call* para sair da fila de processos prontos e aguardar término do filho

Criação de Processos

```
#include <stdio.h>
void main(int arc, char *argv[ ]) {
int pid;
    pid = fork( );    /* bifurcação em outro processo */
    if (pid < 0)      {    /* ocorreu um erro */
        fprintf(stderr,“Fork falhou”);
        exit(-1);
    }
    else if (pid == 0) {    /* processo filho */
        execlp(“/bin/lS”,“lS”,NULL);
    }
    else { /* processo pai */
        wait (NULL);    /* pai espera filho concluir */
        printf(“Filho concluiu”);
        exit(0);
    }
}
```

Término do Processo

- ▶ **Processo executa última instrução**
 - ▶ *System Call* para terminar execução (*exit*)
 - ▶ Dados podem ser enviados ao Pai (*wait*)
 - ▶ Recursos são desalocados pelo SO
 - ▶ Memória, arquivos, *buffers* E/S
 - Armazenados no BCP

Término do Processo

- ▶ Pai pode terminar execução de Filho
 - ▶ *System Call abort*
 - ▶ Filho excedeu uso de recursos
 - ▶ Tarefa do Filho não é mais necessária
 - ▶ Pai está encerrando
 - ▶ Alguns SOs não permitem que os Filhos continuem se o Pai encerrou
 - ▶ Término em cascata

Término do Processo

- ▶ **Unix**
 - ▶ Comando *kill*
 - ▶ Necessário conhecer o PID
 - ▶ Término do Pai
 - ▶ Filhos “adotados” pelo INIT