Sistemas Operacionais I

Modelo de Processo

Multiprogramação de quatro programas

4 processos sequenciais independentes

Somente um programa está ativo

**Tempo de Descompactador**

Baixa a resolução, menos pontos, tráfego menor

Principais elementos que levam à criação de processos:

1- Início de sistema

2- Execução de chamada ao sistema de criação de processos

3- Solicitação do usuário para criar um novo processo

4- início de um job (processo) em lote

**Término de Processos**

1. saída normal (voluntária)

2. saída por erro (voluntário) – de programação

3. erro fatal – tenta escrever na memoria e recebe erro. Não está no nosso controle

4. cancelamento por outro processo (involuntária) – gerenciador de tarefas, por exemplo.

Processo que chama processo 🡪 processo pai

Processo chamado 🡪 processo filho

* Windows não trata hierarquia de processos!!!! Todos os processos são processos pai.

Pronto 🡪 Rodando 🡪

Bloqueado

**27/08/19 – THREADS**

Escalonamento 🡪 cada fluxo de execução é uma thread.

Monothread 🡪 1 processo (fluxo de execução) por vez

Processador Multithread, S.O multithread,

Sem estes itens, não é possível rodar multithread.

Cada thread do usuário é gerenciada por uma thread do núcleo. Isso demanda muito recurso, mas é muito mais fácil em termos de programação. Se uma thread morre, a linha inteira morre também.

Se uma thread morre, procura outra para executar (multithread) (implementar é mais complicado, mas é bem mais eficiente).

Modelo 1-N o Kernel executa todos os threads  
não tem um sistema operacional específico.

Sem o multithreading, o sw não acompanha o hardware do jeito que gostaria

Startar uma thread é MUITO mais leve do que startar um processo inteiro.

Diagrama de Escalonamento de Processos

Pronto – esperando – executando

Dentro do contexto do processo, posso priorizar algumas threads

Starvation

Paralelismo x Concorrência

Threads Daemon

Encosto –

Daemon tools