Sistemas Operacionais

BSI / UAB – 2013
Hélio Crestana Guardia

Visão do SO

- **SO**: camada de **software**, executado diretamente sobre o *hardware* (físico ou virtual)
- Permite que hardware seja usado de forma **eficiente** e **segura**

Visões do Sistema Operacional:

- **Máquina abstrata:**
  - SO esconde detalhes da operação e do uso do hardware
  - SO oferece serviços que simplificam o trabalho do programador
- **Gerente de recursos**:
  - SO controla o funcionamento dos recursos (processador, discos, rede, ...)
  - SO provê compartilhamento seguro e eficiente do uso dos recursos

Processos

- Programas em execução são tratados como processos
- SO usa estrutura de dados para manter informações dos processos:
  - Bloco de controle de processo (*Process Control Block*)
- Informações sobre processos:
  - Identificador do processo (PID), prioridades, credenciais de usuário, permissões, limites de uso de recurso e contabilizações, ponteiros para área de memória contendo o código, ponteiros para área de memória contendo os dados (variáveis estáticas, alocação dinâmica e pilha), informações sobre arquivos abertos, estruturas para tratamento de sinais, mecanismos de comunicação e outros recursos.
  - Pilha de execução
  - Contexto: estado dos registradores do hardware
  - Tabela de páginas
  - Estado de execução: pronto, bloqueado, terminado, …

Threads

- Uma *thread* é uma linha de execução de um processo
- Processos têm ao menos uma *thread*, associada à função *main*() (para programas em C)
- Processos podem ter várias *threads*
- *Threads* de um processo **compartilham** suas áreas de memória
  - Código é o mesmo, embora cada *thread* execute uma função diferente
  - Área de dados e variáveis são as mesmas para o processo e suas *threads*
  - Cada *thread* tem sua própria pilha
  - Cada *thread* tem sua própria cópia do contexto do hardware

Processos X *threads*

- Criação
- Estruturas de controle
- Compartilhamento de informações
- Código que executam: funções [diferentes ou várias instâncias da mesma] dentro do mesmo código

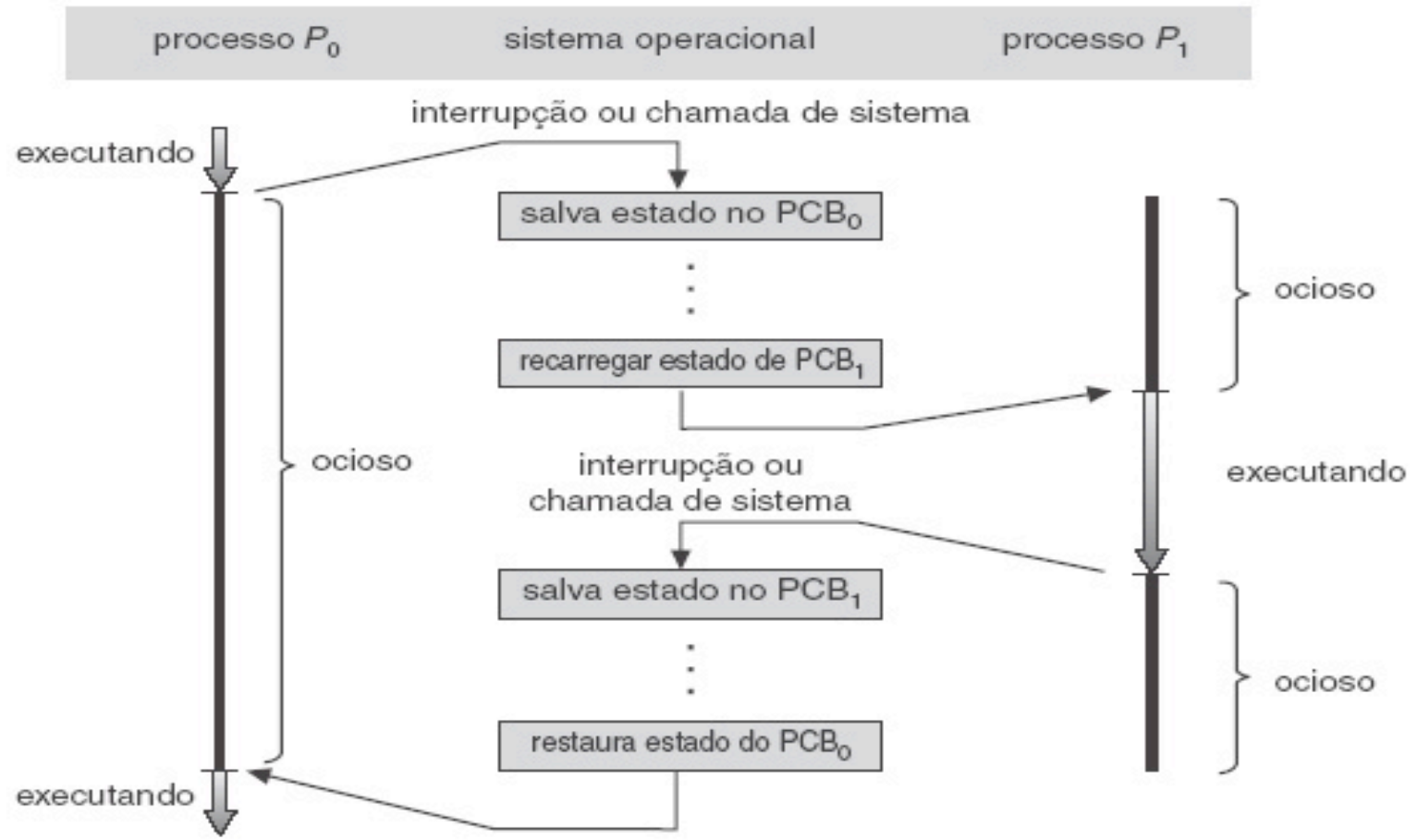
Múltiplos processos

- ***Multitasking***: capacidade do SO de ter mais de um processos (e/ou *threads*) em execução ao mesmo tempo
  - Num dado instante, o número de processos que pode estar tendo suas instruções executadas é limitado ao número de processadores
  - Sistemas Operacionais conseguem salvar contextos de processos (e *threads*) e restaurá-los para continuação da execução posteriormente
  - Execução do processo interrompido pode prosseguir como se ela não tivesse sido interrompida.

Tipos de processos

- CPU Bound: predominância de instruções a executar
- I/O Bound: predominância de operações de Entrada e Saída de dados

Troca de contexto

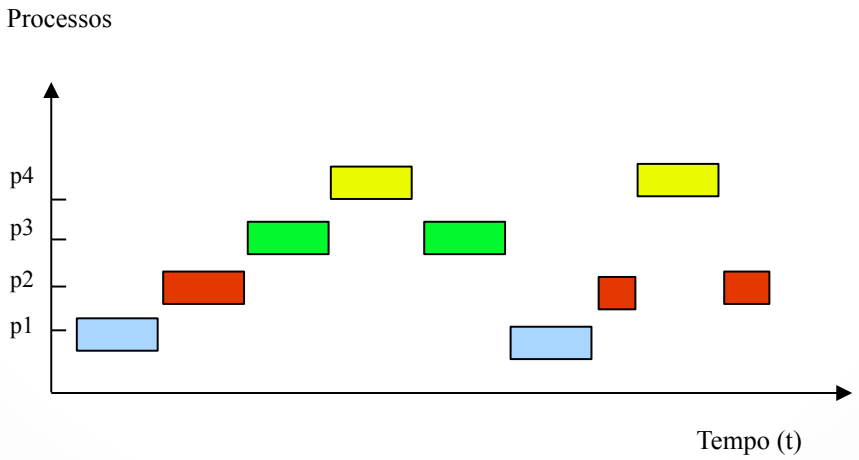


Escalonamento

- Ao compartilhar o uso do processador, SO precisa decidir qual processo a executar quando o processo atual deixa a UCP
- Diferentes políticas podem ser empregadas
- Critérios:
  - Balanceamento do uso
  - Justiça
  - Atendimento a prioridades
  - Rapidez
  - Maximização do número de processos concluídos

Escalonamento

- Algoritmos:
  - FIFO
  - Round Robin (Lista circular)
  - Múltiplas filas de prioridade
  - ...



Interrupções

- Interrupção é um mecanismo que notifica o processador que uma condição excepcional (evento) ocorreu
- Uso de interrupções evita que o SO tenha que ficar periodicamente interagindo com controlador de dispositivos para ver se é preciso ler ou transferir dados
- Tipos de interrupção:
  - **Externa**: gerada por outro dispositivo, como o controlador de disco, interface de rede ou *timer* programável
  - ***Exceção***: gerada em decorrência da execução de instruções:
    - Falta de página
    - Divisão por 0
    - Overflow
    - Acesso inválido à memória
    - Operação ilegal
  - **Instrução** de interrupção: usada para chamada dos serviços do SO

Tratamento de Interrupções

- Ao receber sinal de interrupção ou exceção, o **processador**:
  - Salva na pilha o valor dos *flags* e o conteúdo do registrador que indica a próxima instrução a executar
  - Usa número da interrupção ou exceção para localizar endereço da rotina de tratamento
    - Endereços das rotinas são mantidos numa área de memória (vetor de interrupções)
  - Desvia execução para a rotina de tratamento
    - Endereço da rotina apropriada é carregado no registrador de instruções (PC)

- Qual é o papel do SO no tratamento de interrupção ou exceção?
  - SO cria rotinas de tratamento e as coloca na memória
  - SO preenche vetor de interrupções com os endereços de suas rotinas

- Ação típica de uma rotina de tratamento:
  - Salva registradores que serão usados no código de tratamento da INT
  - Trata evento

  ou
  - Presta serviço solicitado (se INT foi usada para chamada de sistema)
  - Se processo atual precisa **aguardar** condição ou se processo mais **prioritário** ficou **pronto** com o tratamento da INT:
    - Contexto do processo atual é salvo
    - Escalonador é executado para escolher processo a executar
    - Contexto do processo selecionado é restaurado
    - SO devolve libera o uso do processador, que é desviado para a execução do processo selecionado

- Resultado: SO sempre retoma o controle quando uma interrupção ocorre

Chamadas de Sistema

Serviços oferecidos pelo SO para os programas:

- Simplificam a programação
- Proveem acesso seguro aos recursos
- Promovem compartilhamento ordenado e eficiente dos recursos

Acesso:

- Instrução de interrupção *ou*
- Instrução específica de chamada: *syscall* / *sysenter*
- Código do SO é executado com maior nível de privilégio do processador (*kernel mode* / anel de privilégio 0)
- Ao executar com privilégio, SO pode executar instruções restritas de acesso ao hardware

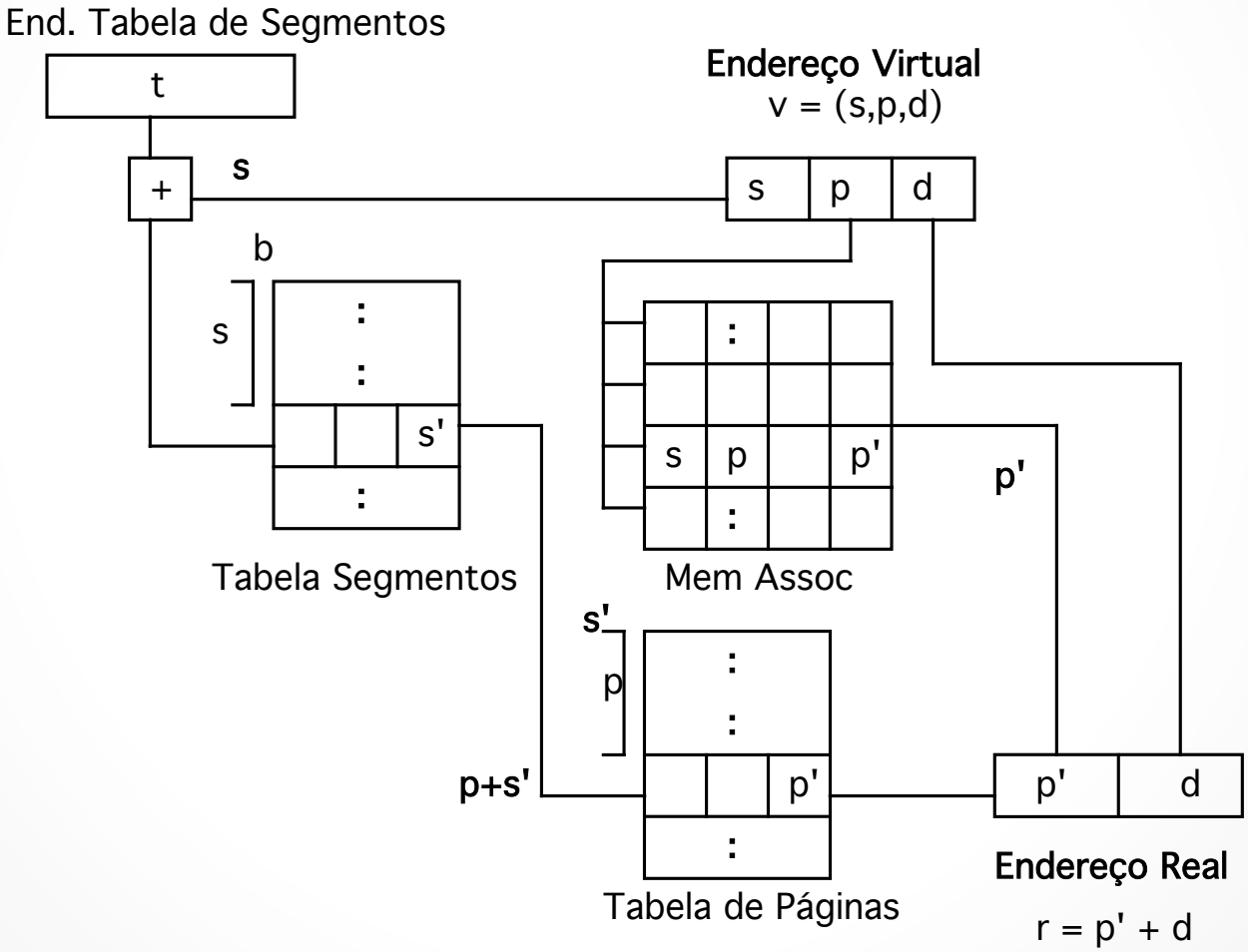
Estratégias do SO

- Para alternar o uso do(s) processador(es) entre vários processos, SO usa **multiprogramação** e **fatias de tempo**
- **Multiprogramação**: sobreposição de serviços de entrada e saída de dados dos processos solicitantes com a execução de código de outro(s) processo pronto
- **Fatias de tempo**: processos não executam sem parar até acabar.
  - Atribuição do uso do processador é feita por períodos
  - Dispositivo de timer presente na placa mãe gera **interrupções periódicas**, programas pelo SO
  - Rotina de tratamento das interrupções do ***timer*** ativa troca de contexto

Gerenciamento de memória

- Espaço de endereçamento dos processos é virtual
- Espaço de memória é organizado e gerenciado pelo SO, criando uma tabela de páginas para cada processo
- Divisão em páginas, contando com o auxílio do hardware
- SO mantém tabela de páginas para cada processo
- Processador (MMU) usa tabela de páginas para tradução dos endereços virtuais em endereços reais
- *Threads* de um processo compartilham suas áreas de memória, portanto usam a mesma tabela de páginas.

Memória virtual

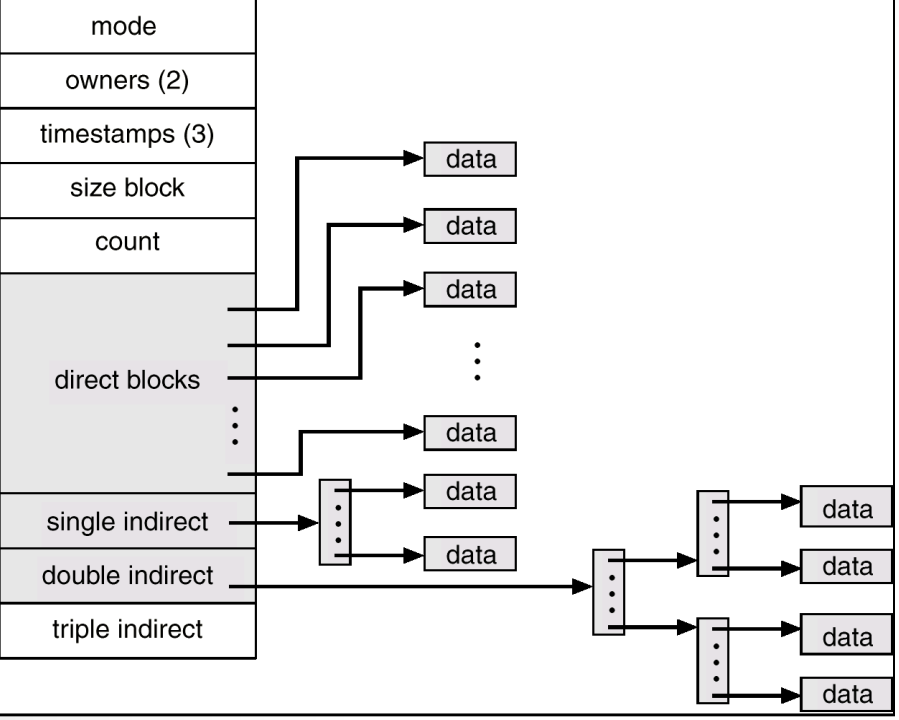


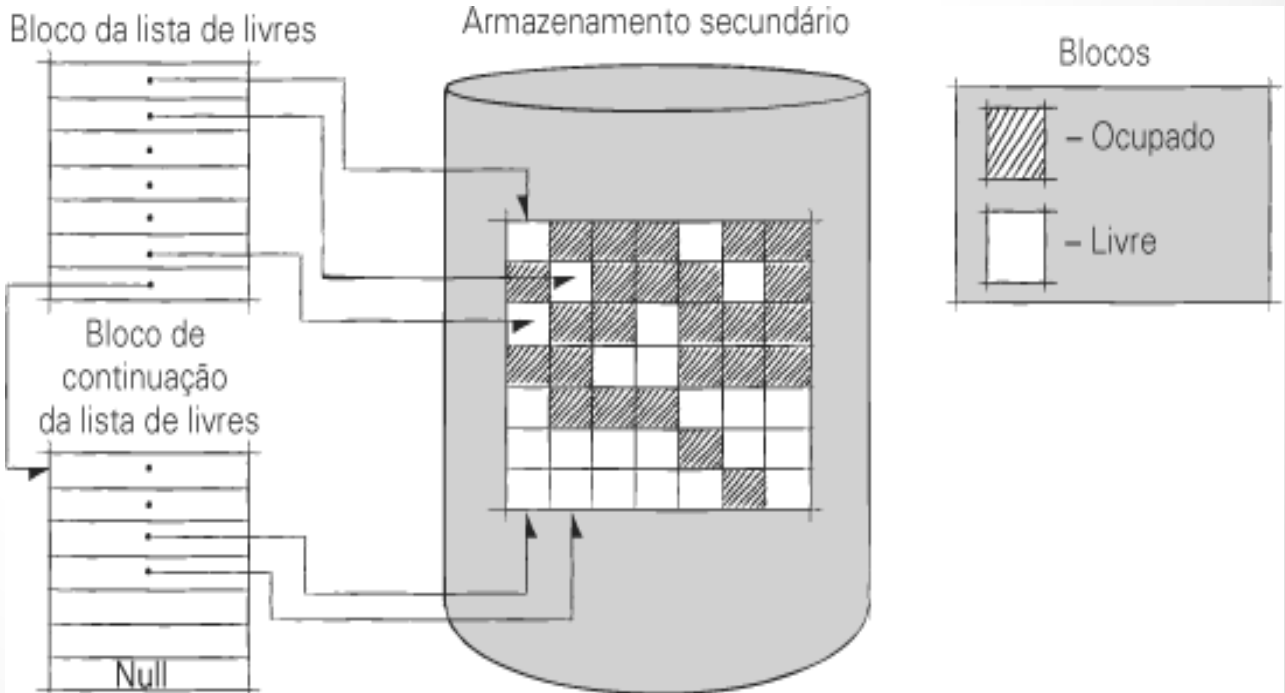
Memória virtual

- Páginas dos programas são carregadas na memória por demanda
- Programas não precisam estar presentes na memória o tempo todo
- Área de *Swap* no disco pode ser usada para auxiliar no armazenamento das páginas não residentes
- Diferentes políticas podem ser usadas para substituir páginas quando há falta de espaço: FIFO, LRU, NRU, …
- Aspectos:
  - *Working set* (conjunto de páginas de trabalho)
  - *Trashing*

Sistemas de Arquivo

- Armazenamento persistente em disco: arquivos e diretórios
- Discos: dispositivos de bloco
  - Requer organização dos blocos
  - Transferências ocorrem em blocos
- SO mantém relação de blocos disponíveis e dos espaços ocupados por diretórios e arquivos
- Formatação: determinação dos blocos existentes e organização de uma estrutura que indica blocos livres e ocupados
- Arquivos: conjuntos de blocos
  - Diferentes formas de organização: listas ou estruturas (i-nodes)
- Operações:
  - Ler, gravar, apagar, renomear, remover, manipular diretórios, …


mode
owners (2)
timestamps (3)
size block
count
direct blocks
single indirect
double indirect
triple indirect
data
data
data
data
data
data
data
data
data
data


Bloco da lista de livres
Armazenamento secundário
Blocos
– Ocupado
– Livre
Bloco de
continuação
da lista de livres
Null

Sistemas de Arquivo

- Transferências de/para uma unidade de armazenamento (disco) é feita em blocos de tamanho fixo (múltiplos de 512 bytes)
- Processos fazem requisição em tamanhos variados
- SO lê blocos, copiando-os para *buffers*, e repassa volumes solicitados aos processos
  - Próximas leituras podem ser atendidas direto dos buffers
- SO gera blocos com dados gravados pelos processos
  - SO pode retardar a escrita, esperando agrupar dados
  - Possíveis inconsistências no sistema de arquivos de unidades removíveis

Sistema E/S

- SO gerencia uso dos dispositivos de E/S
  - Emite comandos para os dispositivos
  - Atende interrupções geradas pelos dispositivos
  - Trata erros nas operações desses dispositivos
  - Provê uma interface para utilização dos dispositivos; se possível a mesma para todos eles, de forma que a sintaxe das operações seja independente dos dispositivos.
- Controladores de dispositivo (*device drivers*) fornecem interface aos dispositivos físicos
- Interação com controladores:
  - Leitura e escrita via barramento: IN e OUT
  - DMA
  - Interrupções

Sistema E/S

- Organização do software de E/S
  - Tratadores de interrupção
  - Controladores de dispositivos
  - *Software* de E/S independente de dispositivo
  - *Software* de E/S ao nível do usuário

Dúvidas?

Boa prova!