

Arquitetura e Organização de Computadores

Capítulo 8

Suporte do sistema operacional

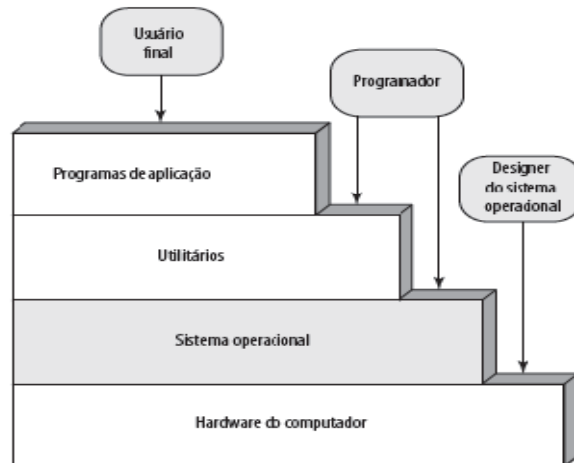
slide 1

© 2010 Pearson Prentice Hall. Todos os direitos reservados.

Objetivos e funções

- Conveniência:
 - Tornar o computador mais fácil de usar.
- Eficiência:
 - Permitir o melhor uso dos recursos do computador.
- Funções:
 - Escalonamento de Processos (qual processo roda em que momento?)
 - Gerenciamento de Memória (principal e virtual)

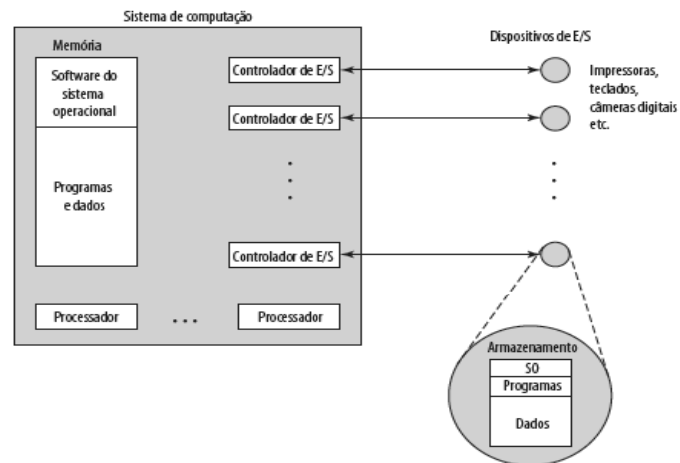
Camadas e visões de um sistema de computação



Serviços do sistema operacional

- Criação de programas (editores, depuradores, ...).
- Execução de programas .
- Acesso aos dispositivos de E/S (um é diferente do outro e precisam ser inicializados).
- Acesso controlado aos arquivos (formato e mecanismos de proteção).
- Acesso ao sistema (proteção de recursos, dados e solução de conflitos de disputa pelo recurso).
- Detecção e resposta a erros (erros de hardware e de software).
- Contabilidade (accounting).

S/O como gerenciador de recursos



Tipos de sistema operacional

- Interativo.
- Em lote (*batch*). Raros hoje.
- Um programa por vez (uniprogramação).
- Multiprogramação (multitarefa).

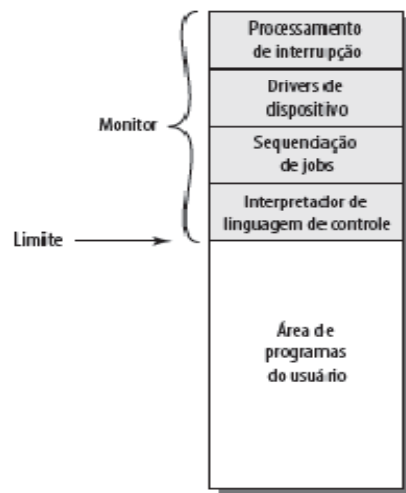
Primeiros sistemas

- Final da década de 1940 a meados da década de 1950.
- Sem sistema operacional.
- Programas interagem diretamente com o hardware.
- Dois problemas principais:
 - Escalonamento inadequado.
 - Tempo de preparação longo.

Sistemas em lote simples

- Programa monitor residente.
- Usuários submetem jobs ao operador.
- Operador coloca jobs em lotes.
- Monitor controla sequência de eventos para processar lote.
- Quando um job termina, o controle retorna ao monitor, que lê próximo job.
- Monitor trata do escalonamento.

Layout de memória para um monitor residente



Job Control Language

- Instruções ao monitor.
- Normalmente indicada por \$.
- P.e.:
 - \$JOB
 - \$FTN
 - ... Algumas instruções em Fortran.
 - \$LOAD
 - \$RUN
 - ... Alguns dados.
 - \$END

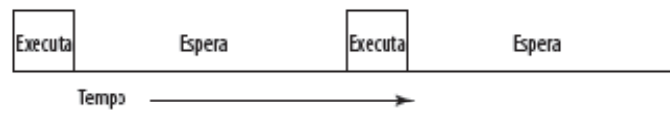
Recursos desejáveis do hardware

- Proteção da memória:
 - Para proteger o monitor.
- Temporizador:
 - Para impedir que o job monopolize o sistema.
- Instruções privilegiadas:
 - Executadas apenas pelo monitor.
 - P.e., E/S.
- Interrupções:
 - Permitem abdicar e retomar o controle.

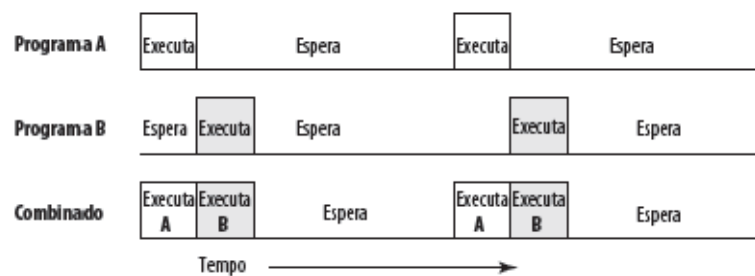
Sistemas em lote multiprogramados

- Dispositivos de E/S muito lentos.
- Quando um programa está esperando E/S, outro pode usar a CPU.

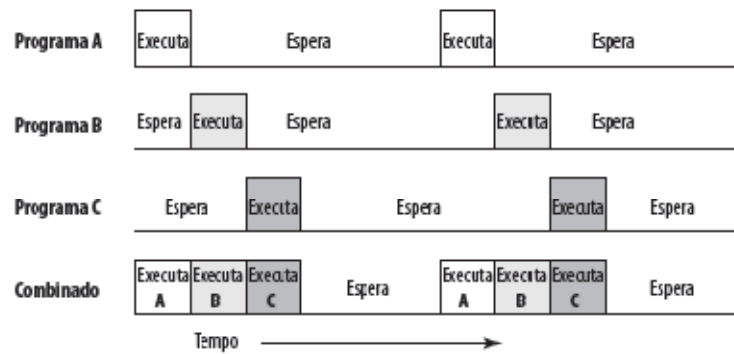
Único programa



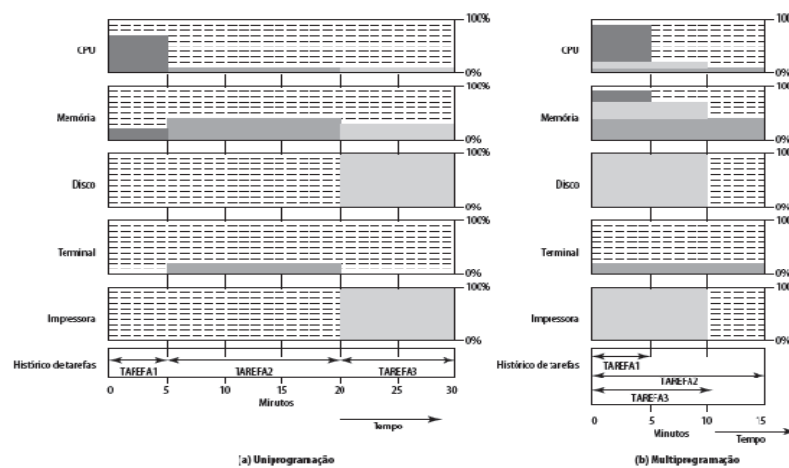
Multiprogramação com dois programas



Multiprogramação com três programas



Histogramas de utilização



Sistemas de tempo compartilhado

- Permitem que os usuários interajam diretamente com o computador.
 - Ou seja, interativos.
- Multiprogramação permite que uma série de usuários interajam com o computador.

Escalonamento

- Chave para multiprogramação.
- Longo prazo.
- Médio prazo.
- Curto prazo.
- E/S.

Escalonamento a longo prazo

- Determina quais programas são submetidos para processamento.
- Ou seja, **controla o grau de multiprogramação**.
- Uma vez submetido, um job torna-se um processo para o escalonador a curto prazo (ou torna-se um job não carregado na memória para o escalonador a médio prazo).

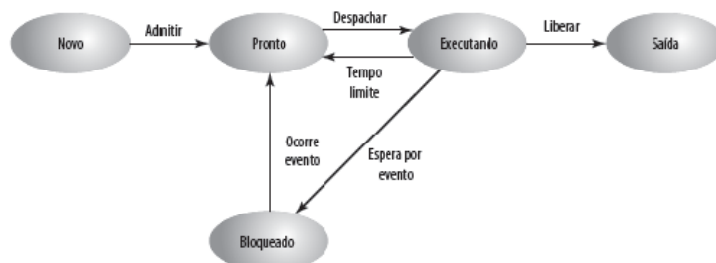
Escalonamento a médio prazo

- Parte da função de troca de processo (***swapping***).
- Swapping é a troca de um processo que está em memória (p.e. esperando E/S), por um processo que está no disco em fila intermediária. A execução continua com o processo recém-chegado.
- Normalmente baseado na necessidade de gerenciar a multiprogramação.
- Se não há memória virtual, o gerenciamento de memória também é um ponto.

Escalonamento a curto prazo

- Despachante (*dispatcher*) - é o escalonador de curto prazo.
- Decisões de nível mais baixo de qual tarefa executar em seguida.
- Ou seja, **qual tarefa realmente usa o processador** no próximo intervalo de tempo.

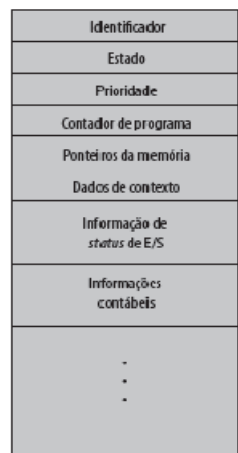
Modelo de processo com cinco estados



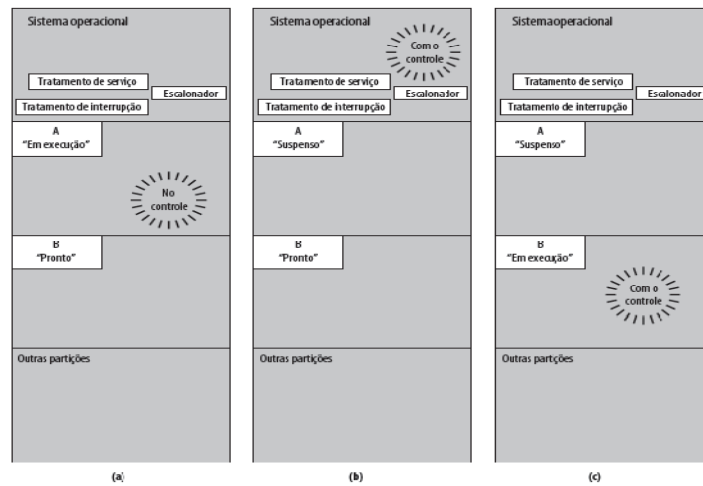
Bloco de controle de processo (representação de um processo pelo SO)

- Identificador.
- Estado.
- Prioridade.
- Contador de programa.
- Ponteiros de memória (locais inicial e final do processo na memória).
- Dados de contexto.
- Status de E/S.
- Informações contábeis (tempo de processador, tempo de clock usado, limites de tempo, etc).

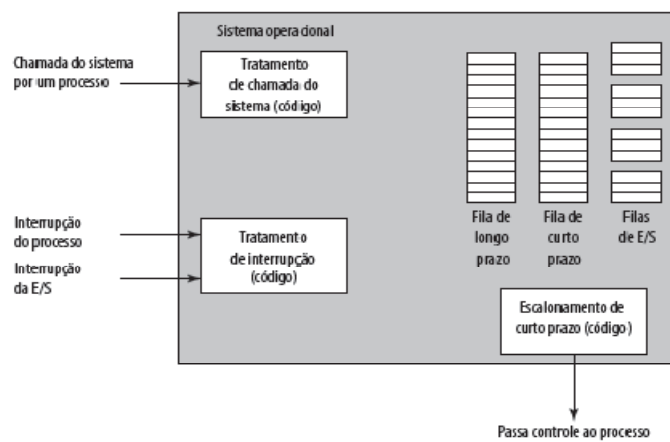
Diagrama do bloco de controle de processo



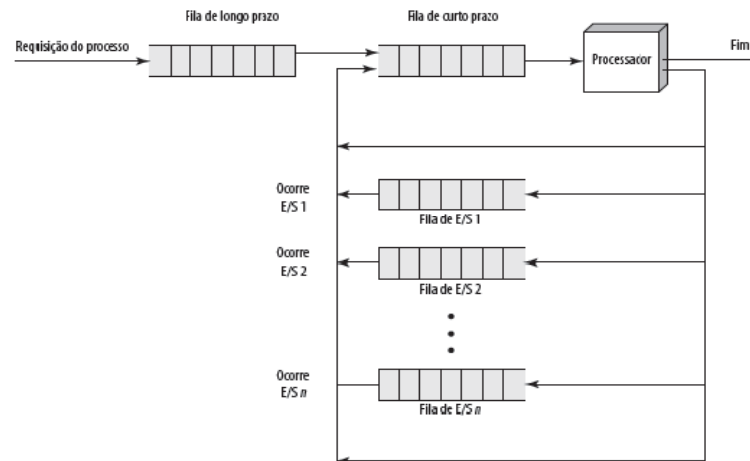
Exemplo de escalonamento



Principais elementos do sistema operacional



Escalonamento de processador



Gerenciamento de memória

- Uniprogramação:
 - Memória dividida em duas.
 - Uma para sistema operacional (monitor).
 - Uma para programa atualmente em execução.
- Multiprogramação:
 - Parte do "usuário" é subdividida e compartilhada entre processos ativos.

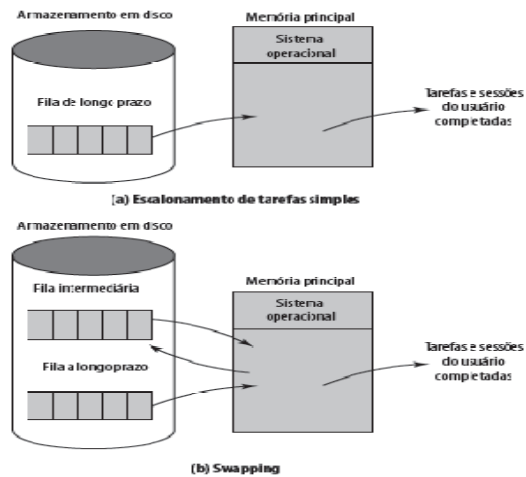
Swapping (troca de processos)

- Problema: E/S é tão lenta, em comparação com a CPU, que até mesmo em sistema de multiprogramação a CPU pode estar ociosa na maior parte do tempo.
- Soluções:
 - Aumentar memória principal.
 - Cara.
 - Leva a programas maiores.
 - Swapping (troca de processos).

O que é swapping?

- Fila a longo prazo dos processos armazenados no disco.
- Processos “trocados” para a memória quando existe espaço disponível.
- Quando um processo termina de executar, ele é movido para fora da memória principal.
- Se nenhum dos processos na memória estiver pronto (ou seja, toda a E/S bloqueada).
 - Retirar um processo bloqueado para a fila intermediária.
 - Entra com um processo pronto ou um novo processo.
 - Mas o swapping é um processo de E/S...

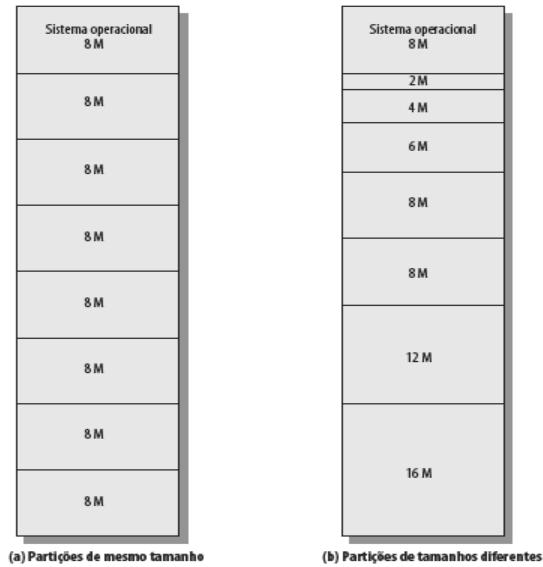
Uso do swapping



Particionamento

- Dividir a memória em seções para alocar processos (incluindo sistema operacional).
- Partições de tamanho fixo:
 - Podem não ser do mesmo tamanho.
 - Processo é encaixado no menor espaço que o poderá conter.
 - Alguma memória desperdiçada.**
 - Leva a partições de tamanho variável.

Particionamento fixo

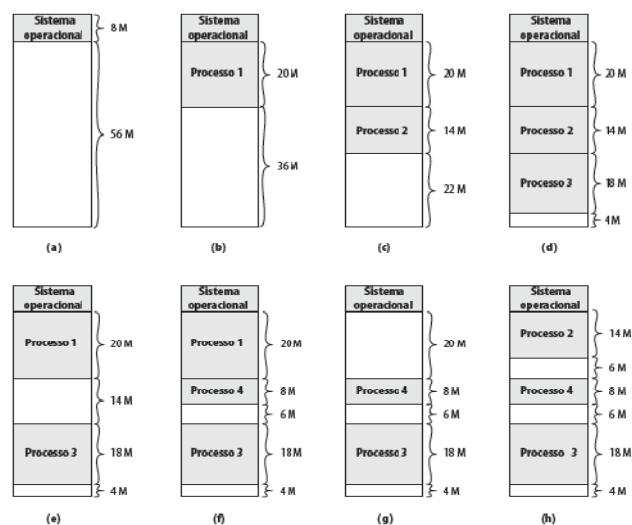


Partições de tamanho variável

- Alocam exatamente a memória requisitada a um processo.
- Isso leva a um buraco no final da memória, muito pequeno para ser usado.
 - Apenas um buraco pequeno – menos desperdício.
- Quando todos os processos estão bloqueados, retira um processo e traz outro.
- Novo processo pode ser menor que o processo removido.
- **Outro buraco.**

- Por fim terá muitos buracos (**fragmentação**).
- Soluções:
 - Aglutinação – juntar buracos adjacentes em um grande buraco.
 - Compactação** – de vez em quando, percorre a memória e move todos os buracos para um bloco livre (**desfragmentação de disco**).

Efeito do particionamento dinâmico



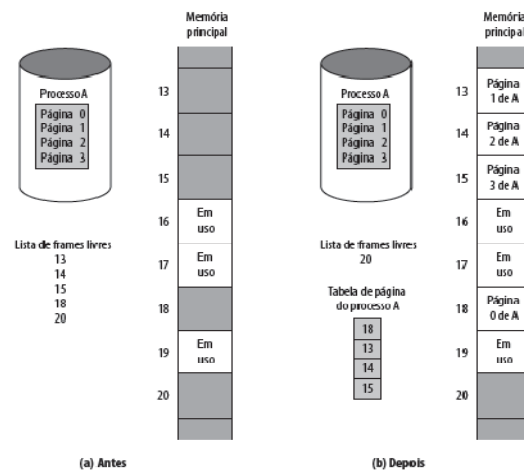
Relocação

- Nenhuma garantia de que o processo será carregado no mesmo local na memória.
- **Instruções contêm endereços:**
 - 1. Localizações dos dados.
 - 2. Endereços para instruções (desvio).
- Endereço lógico – relativo ao início do programa.
- Endereço físico – local real na memória (desta vez).
- Conversão automática usando endereço de base.

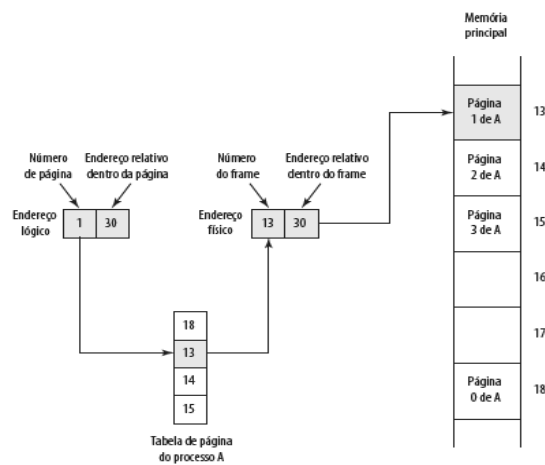
Paginação

- Memória dividida em pedaços pequenos de mesmo tamanho – **frames de página**.
- Divide programas (processos) em pedaços pequenos de mesmo tamanho – **páginas**.
- Aloca o número exigido de *frames* de página a um processo.
- SO mantém lista de **frames livres**.
- Um processo não exige *frames* de página contíguos.
- SO usa **tabela de página** para cada processo. Tabela mostra o local para cada página no processo.

Alocação de *frames* livres



Endereços lógicos e físicos – paginação



Memória virtual

- Paginação por demanda:
 - Não exige todas as páginas de um processo na memória.
 - Traz páginas conforme a necessidade (p.e. uma subrotina e alguns dados, mas de uso frequente).
- Falta de página:
 - Página exigida não está na memória (Interrupção será gerada).
 - Sistema operacional deve trazer a página requisitada.
 - Pode ter que retirar uma página para criar espaço.
 - Seleciona página para remover com base na história recente.

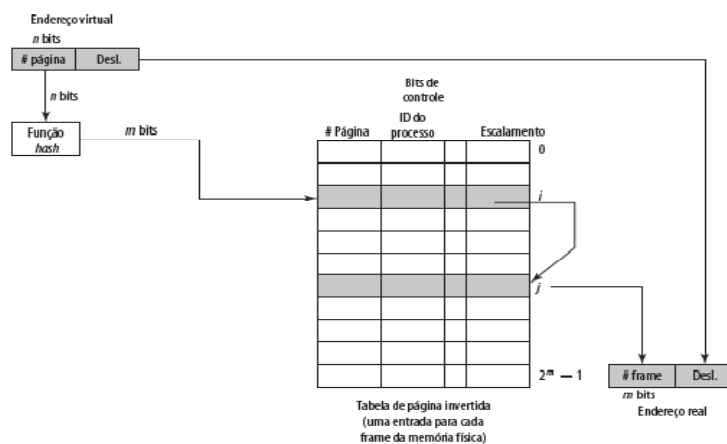
Thrashing

- Muitos processos em muito pouca memória.
- Sistema operacional gasta todo o seu tempo trocando páginas.
- Pouco ou nenhum trabalho real é feito.
- **Luz do disco acesa o tempo todo.**
- Soluções:
 - Bons algoritmos de substituição de página.
 - Reduzir número de processos em execução.
 - Colocar mais memória.

Bônus

- Não precisamos de um processo inteiro na memória para que ele seja executado.
- Podemos trocar páginas conforme a necessidade.
- Assim, podemos agora executar processos que são maiores que a memória total disponível!
- Memória principal é denominada memória real.
- Usuário/programador vê memória muito maior – aquela que está alocada no disco – **Memória Virtual**.

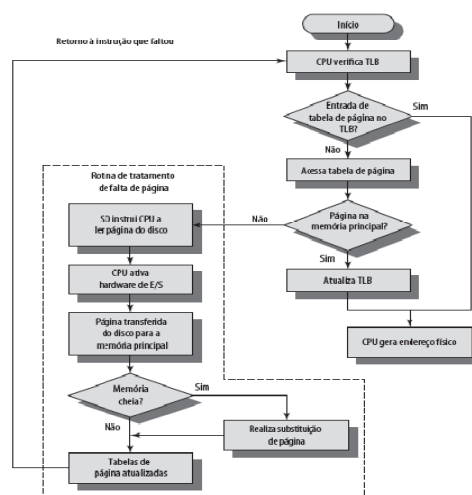
Estrutura de tabela de página invertida



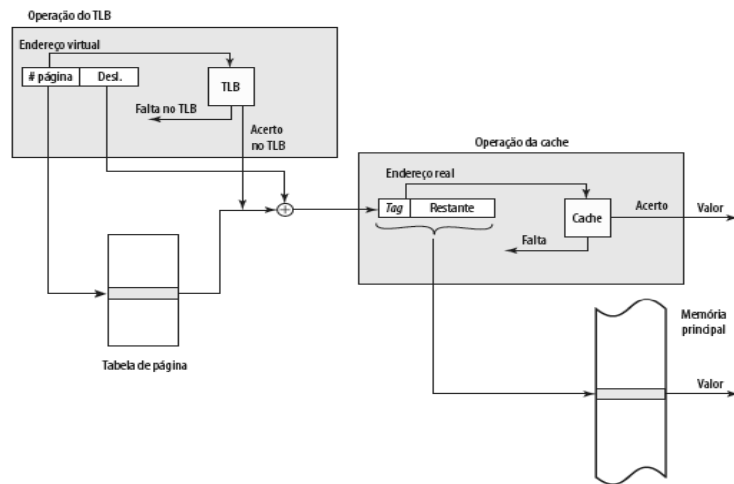
Translation Lookaside Buffer

- Cada referência à memória virtual causa dois acessos à memória física:
 - Buscar entrada da tabela de página apropriada.
 - Buscar dados desejados.
- Usar **cache especial para tabela de página**.
 - TLB.

Operação do TLB



TLB e operação da cache



Segmentação

- Paginação não é (normalmente) visível ao programador.
- **Segmentação é visível ao programador.**
- Normalmente, diferentes segmentos alocados a programa e dados.
- Pode ser uma série de segmentos de programa e dados.

Vantagens da segmentação

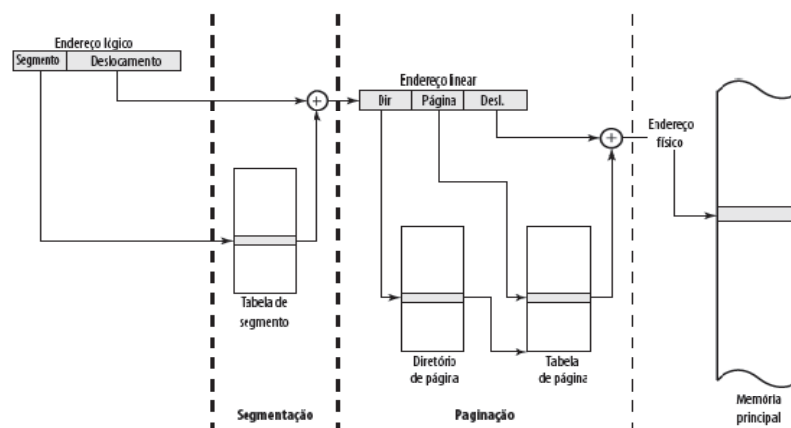
- Simplifica o tratamento de estruturas de dados que crescem.
- Permite que programas sejam alterados e recompilados independentemente, sem nova link-edição e recarregamento.
- Permite compartilhamento entre processos.
- Serve para proteção.
- Alguns sistemas combinam segmentação com paginação.

Pentium II

- Hardware para segmentação e paginação:
- 1. Não segmentada, não paginada.
 - endereço virtual = endereço físico.
 - Baixa complexidade.
 - Alto desempenho.
- 2. Não segmentada, paginada.
 - Memória vista como espaço de endereço linear paginado.
 - Proteção e gerenciamento via paginação.
 - Ex. Berkeley UNIX.

- 3. Não paginada, segmentada:
 - Coleção de espaços de endereços lógicos.
 - Proteção em nível de único byte.
 - Tabela de tradução necessária está no chip quando segmento está na memória.
- 4. Paginada, segmentada:
 - Segmentação usada para definir partições de memória lógicas sujeitas ao controle de acesso.
 - Paginação gerencia alocação de memória dentro das partições.
 - Ex. Unix System V.

Mecanismo de tradução de endereço de memória no Pentium II



Segmentação no Pentium II

- Cada endereço virtual é um segmento de 16 bits e um offset de 32 bits.
- 2 bits do segmento constituem mecanismo de proteção.
- 14 bits especificam segmento.
- Memória virtual não segmentada $2^{32} = 4$ GBytes
- Segmentada $2^{46} = 64$ Tbytes.
 - Pode ser maior – depende do processo ativo.
 - Metade (8K segmentos de 4GB) é global.
 - Metade é local e distinta para cada processo.

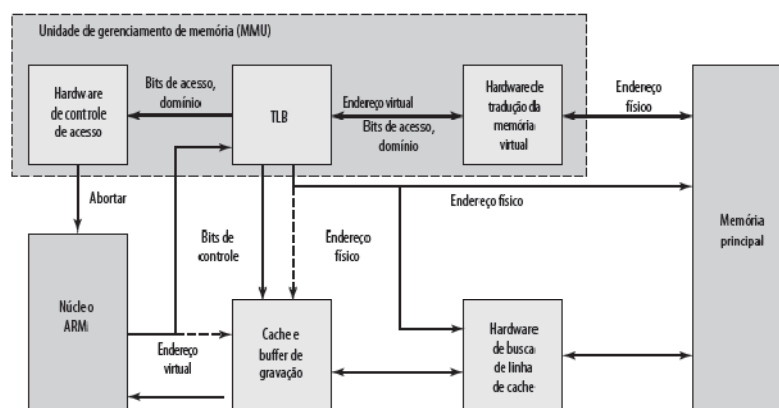
Proteção no Pentium II

- Os bits de proteção geral têm 4 níveis de privilégio:
 - 0 mais protegido; 3, o menos.
 - Uso de níveis depende do software.
 - Geralmente, **nível 3 para aplicações, nível 1 para SO e nível 0 para kernel** (nível 2 não usado).
 - Nível 2 pode ser usado para aplicações que possuem segurança interna, p.e., banco de dados.
 - Algumas instruções só funcionam no nível 0.

Paginação no Pentium II

- Segmentação pode ser desabilitada:
 - Nesse caso, usa espaço de endereços linear.
- Dois níveis de pesquisa na tabela de página:
 - Primeiro, diretório de página.
 - 1024 entradas no máximo.
 - Divide memória linear de 4G em 1024 grupos de página de 4MB.
 - Cada tabela de página tem 1024 entradas correspondentes a 4KB páginas.
 - Pode usar diretório de página para todos os processos, um por processo ou mistura.
 - Diretório de página para processo atual sempre na memória.
 - Usa TLB mantendo 32 entradas da tabela de página.
 - Dois tamanhos de página disponíveis: 4k ou 4M.

Visão geral do sistema de memória do ARM



Gerenciamento de memória do ARM

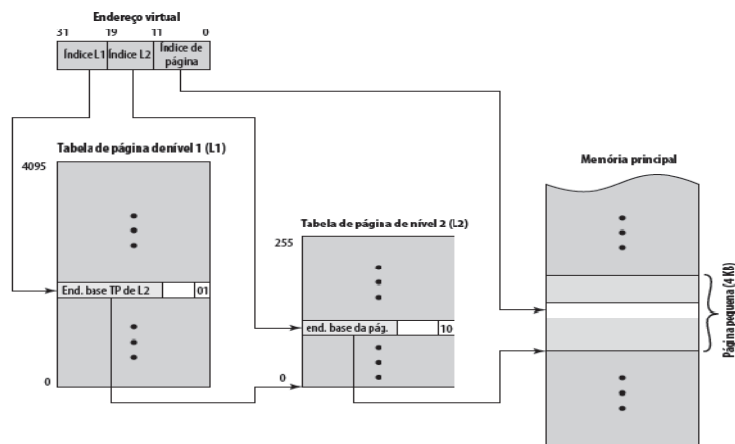
- Tradução de memória virtual:
 - Um ou dois níveis de tabela.
- *Translation Lookaside Buffer* (TLB).
 - Cache de entradas recentes.
 - Se estiver disponível, TLB envia diretamente endereço físico à memória principal.
- Dados trocados entre processador e memória principal via cache.
- Organização lógica da cache.
 - Na falta de cache, ARM fornece endereço diretamente à cache e também ao TLB.
- Organização física da cache:
 - TLB precisa fornecer endereço físico à cache.
- Bits de controle de acesso também nas tabelas de tradução.

Tradução de endereço de memória virtual

- Acesso à memória baseado em seções ou páginas.
- Superseções (opcional):
 - blocos de 16 MB de memória principal.
- Seções:
 - Blocos de 1 MB de memória principal.
- Páginas grandes:
 - Blocos de 64 KB de memória principal.
- Páginas pequenas:
 - Blocos de 4 KB de memória principal.
- Seções e superseções permitem o mapeamento de grande região de memória com única entrada de TLB.
- Mecanismos adicionais de controle de acesso:
 - Páginas pequenas usam subpáginas de 1KB.
 - Páginas grandes usam subpáginas de 16KB.

- Tabela de tradução de dois níveis mantida na memória principal.
 - Tabela de primeiro nível mantém traduções de seção e superseção, e ponteiros para tabelas de segundo nível.
 - Tabelas de segundo nível: Mantêm traduções de páginas grandes e pequenas.
- MMU:
 - Traduz endereços virtuais para físicos.
 - Deriva e verifica permissão de acesso.
 - Após falta de TLB.
- Começa com busca de primeiro nível.
 - Acesso mapeado por seção só requer busca de primeiro nível.
 - Acesso mapeado por página também requer busca de segundo nível.

Tradução de endereço de memória virtual do ARM para páginas pequenas - Diagrama



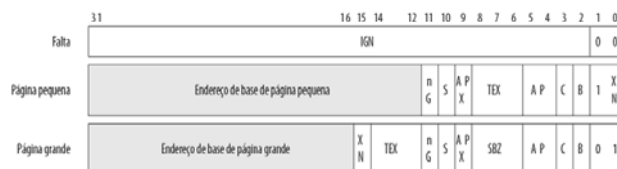
Tradução de endereço de memória virtual do ARM para páginas pequenas

- Única tabela de página L1:
 - 4K entradas de 32 bits.
 - Cada entrada L1 aponta para tabela de página L2.
- Cada tabela de página L2:
 - 255 entradas de 32 bits.
 - Cada entrada L2 aponta para página de 4 KB na memória principal.
- Endereço virtual de 32 bits:
 - 12 bits mais significativos indexam tabela de página L1.
 - Próximos 8 bits indexam tabela de página L2 relevante.
 - 12 bits menos significativos indexam um byte na página de memória principal relevante.
- Procedimento semelhante para páginas grandes.
- Seções e superseção usam apenas tabela de página L1.

Figura 8.24 Formatos de gerenciamento de memória do ARMv6



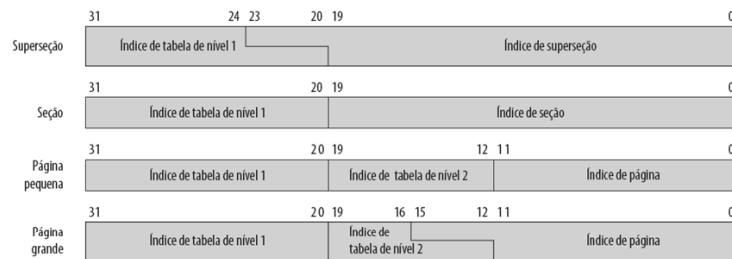
(a) Formatos alternativos de descritor de primeiro nível



(b) Formatos alternativos de descritor de segundo nível

(Continua)

(Continuação)



(c) Formatos de endereço de memória virtual

Parâmetros de gerenciamento de memória do ARM

- **Permissão de acesso (AP), Extensão de permissão de acesso (APX):**
 - Controlam acesso à região de memória correspondente.
 - Acesso sem permissões requeridas geram falta de permissão.
- **Bit Bufferable (B):**
 - Com bits TEX bits, determina como o buffer de escrita é usado.
- **Bit Cacheable (C):**
 - A região da memória pode ser mapeada através de cache?
- **Domínio:**
 - Coleção de regiões da memória.
 - Controle de acesso pode ser aplicado na base do domínio.
- **não Global (nG):**
 - Tradução marcada como global (0), ou específica do processo (1)?

- **Compartilhada (S):**
 - Tradução é para memória não compartilhada (0), ou compartilhada (1)?
- **SBZ:**
 - Deve ser zero (Should Be Zero).
- **Extensão de tipo (TEX):**
 - Junto com bits B e C, controla acessos às caches.
 - Como o buffer de escrita é usado.
 - Se a região da memória é compartilhável:
 - Deve permanecer coerente.
- **Executar nunca (XN):**
 - Região é executável (0) ou não executável (1)?

Formatos de gerenciamento de memória – Tabela L1

- Tabela L1:
 - Entrada descreve como intervalo de endereços virtuais de 1 MB é mapeado.
- Bits [1:0] = 00:
 - Endereços virtuais não mapeados.
 - Tentativas de acesso geram falta de tradução.
- Bits [1:0] = 01:
 - Endereço físico da tabela de página L2 que especifica como são mapeados intervalos de endereços virtuais associados.
- Bits [1:0] = 01 e bit 19 = 0:
 - Descritor de seção.
- Bits [1:0] = 01 e bit 19 = 1:
 - Descritor de superseção.
- Entradas com bits [1:0] = 11:
 - Reservadas.

Tabela L2 – Páginas pequenas e grandes

- Para memória estruturada em páginas.
- Bits de entrada de página L1 [31:10] apontam para uma tabela de página L2.
- Páginas pequenas:
 - Entrada L2 mantém ponteiro de 20 bits para endereço de base da página de 4 KB na memória principal.
- Páginas grandes:
 - Endereço virtual inclui índice de 12 bits para tabela L1 e índice de 8 bits para tabela L2.
 - Páginas grandes de 64 KB possuem parte de índice de página de 16 bits.
 - Sobreposição de quatro bits entre campo de índice de página e campo de índice de tabela L2.
 - Entradas de tabela de página na tabela de página L2 replicadas 16 vezes.
 - Tabela de página L2 reduzida de 256 entradas para 16 se todas se referirem a páginas grandes.
 - Página L2 pode atender mistura de páginas grandes e pequenas.

Tabela L2 – Seções e superseções

- Seções ou superseções:
 - Acesso à tabela de página de um nível.
- Seções:
 - Bits [31:20] da entrada L1 mantém ponteiro de 12 bits para seção de 1 MB.
- Para superseções:
 - L1 bits [31:24] mantém ponteiro de 8 bits para base da seção de 16 MB.
- Replicação da entrada da tabela de página necessária:
 - Em superseções, parte do índice da tabela L1 do endereço virtual sobrepõe 4 bits com parte de índice da superseção do endereço virtual.
 - 16 entradas de tabela de página L1 idênticas.
- Espaço de endereço físico pode ser expandido para oito bits de endereço adicionais:
 - Bits [23:20] e [8:5].
 - Dependente da implementação.
 - Interpretado como estendendo memória física por até $2^8 = 256$.
 - Memória física pode ter até 256 vezes o espaço de memória disponível a cada processo individual.

Controle de acesso

- Região da memória pode ser designada como sem acesso, apenas de leitura ou leitura/escrita.
- Região pode ser designada como apenas para acesso privilegiado (sistemas operacionais).
- Domínio:
 - Coleção de seções e/ou páginas com permissões de acesso particulares.
 - 16 domínios.
 - Múltiplos processos podem usar as mesmas tabelas de tradução enquanto mantêm alguma proteção um do outro.
 - Entrada de tabela de página e entrada de TLB contêm campo de domínio.
 - Campo de dois bits no *Domain Access Control Register* controla acesso a cada domínio.
 - Áreas de memória inteiras podem entrar e sair da memória virtual de modo bastante eficiente.

- Clientes:
 - Devem observar permissões de acesso das seções e/ou páginas individuais no domínio.
- Gerentes:
 - Controlam comportamento do domínio.
 - Seções e páginas no acesso ao domínio.
 - Contornam permissões de acesso para entradas de tabela no domínio.
- Programas podem:
 - Ser clientes de alguns domínios.
 - Ser gerentes de outros domínios.
 - Não ter acesso aos domínios restantes.