

# Sistemas Operacionais

## Módulo 2 – Gerenciamento de Memória

# Gustavo Vilar

- Mini – CV
  - PPF / DPF – Papiloscopista Policial Federal
  - Pós-Graduado em Docência do Ensino Superior – UFRJ
  - Graduado em Ciência da Computação e Processamento de Dados – ASPER/PB
  - Aprovações: PRF 2002, PF 2004, MPU 2010, ABIN 2010

# Gustavo Vilar

- Contatos:

- [gustavopintovilar@gmail.com](mailto:gustavopintovilar@gmail.com)

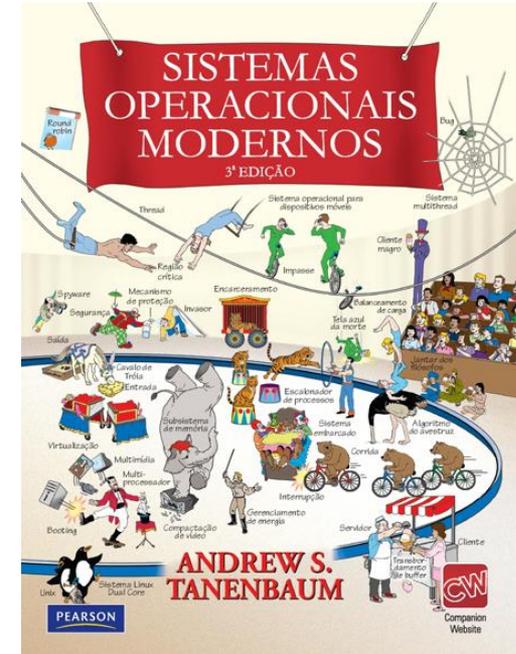
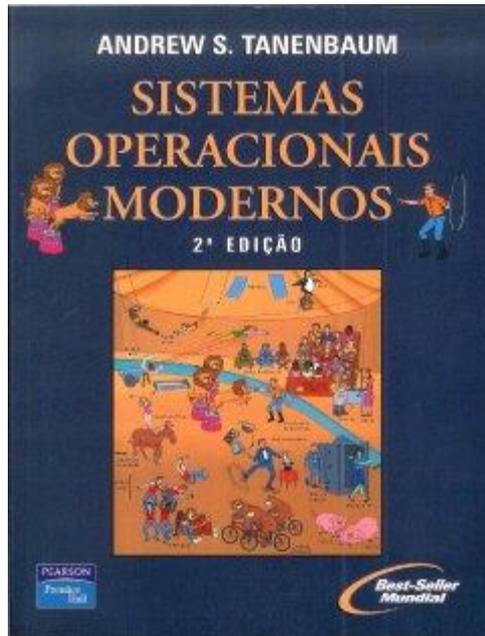
- [p3r1t0f3d3r4l@yahoo.com.br](mailto:p3r1t0f3d3r4l@yahoo.com.br)



# Escopo

- Abordar os assuntos mais recorrentes e com fortes tendências para concursos atuais
- Familiarizar o concursando com os tipos de questões mais frequentes.
- Abordar as metodologias de resolução de questões das principais bancas

# Bibliografia



# Sistemas Operacionais 02 de 03 – Carga Horária

- **12 vídeo aulas (03h54m / 00h19m30s)**
  - Aspectos básicos da memória, conceitos e políticas;
  - Responsabilidades do SO, estratégias e técnicas empregadas;
  - MMU, Monoprogramação, Multiprogramação, Segmentação, Paginação;
  - Controle dos espaços livres, algoritmos de escolha dos espaços livres;
  - Primeira bateria de questões de aprendizagem;
  - Endereçamento virtual, paginação;
  - TLB, paginação, tabela de páginas, fragmentação interna;
  - Segmentação, segmentação paginada, escalonamento;
  - Segunda bateria de questões de aprendizagem;
  - Escalonadores de longo, médio e curto prazo;
  - Memória secundária, estrutura do disco rígido;
  - Terceira bateria de questões de aprendizagem

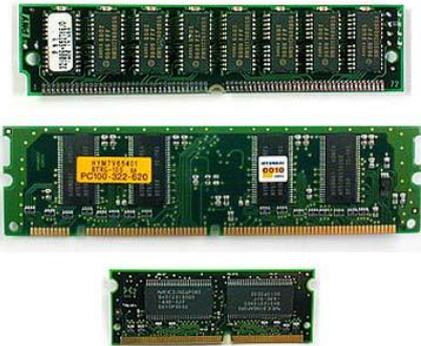


# Sistemas Operacionais

## Gerenciamento de Memória

# Memória principal

- Processo deve estar na RAM para ser executado
- No passado os programas deveriam ser pequenos para caberem totalmente na RAM (que era pequena)
- Técnicas de gerenciamento eram complicadas
- Predominante nos SOs monoprogramados



# Esquema visual



# Política de utilização da memória

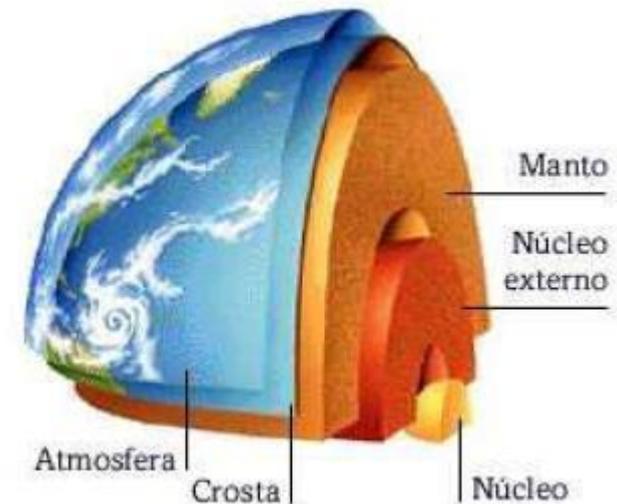
- Manter na RAM a maior quantidade de processos possível
  - realocação (ou recolocação)= assegurar que cada processo tenha o seu próprio espaço de endereçamento, referenciando um registrador base
- Admitir swapping (troca MD)
  - Programas maiores que a RAM
  - Transferência temporária de processos residentes na memória principal para memória secundária
- Garantir proteção às áreas do SO

# Esquema visual



# Conceitos associados

- Overlay
  - Programas eram maiores do que a memória disponível
  - Módulo principal sempre na memória gerenciando os demais módulos (cadastro, impressão, etc...)
  - Divididos em camadas/módulos para execução



# Conceitos associados

- Reentrância
  - Capacidade carregar o código uma única vez e compartilhá-lo por diversos processos/usuários
    - Módulo de impressão
    - DLLs

# Responsabilidades do SO na gerência de memória

- O gerenciamento de memória é feito pelo Sistema Operacional
- Controle das partes, alocação e liberação
- Trocas de processos
  - Memória Virtual
  - Swapping



# Estratégias de gerenciamento de memória

- “Caso os programas não estejam na RAM”
- **Busca**
  - Quando/como transferir um processo, ou parte dele, do disco para RAM?
  - Por demanda
    - Transferência de páginas para memória principal quando são referenciadas
  - Por previsibilidade
    - Busca Antecipada
    - Além da página referenciada, são carregadas outras páginas que também podem (ou não) ser referenciadas futuramente
    - conjunto de trabalho
- **Posicionamento**
  - Local da RAM onde colocar os processos
- **Substituição**
  - Maior problema
  - Qual processo deve sair da RAM para dar lugar a outro
  - De preferência deverá entrar na área livre



# Esquema visual

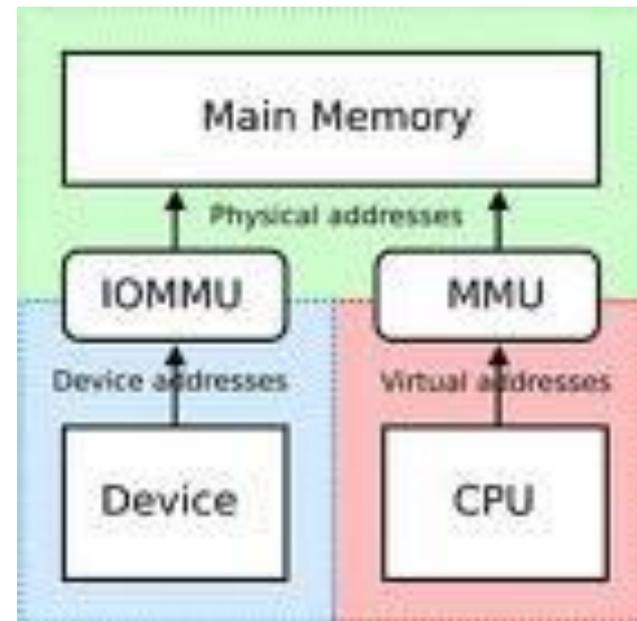


# Técnicas empregadas

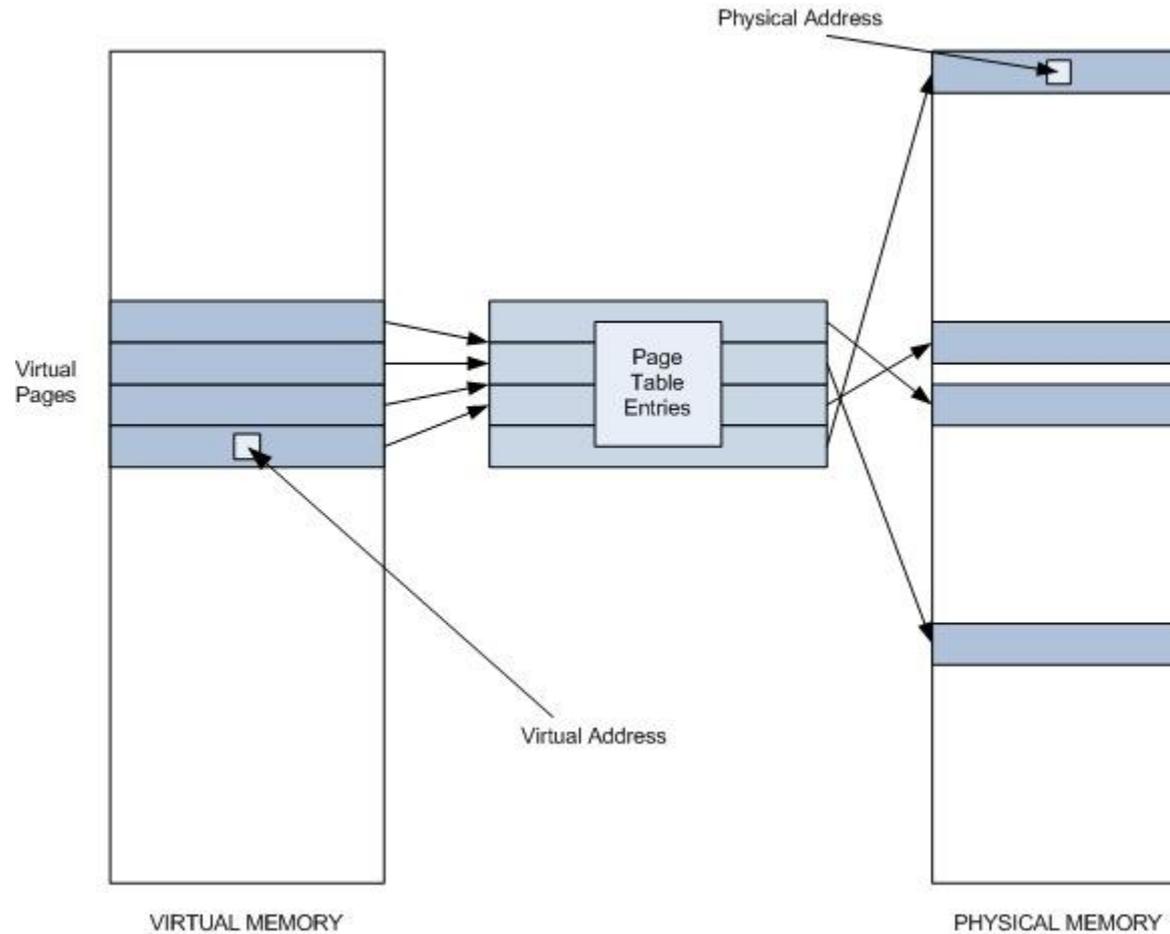
- **Paginação**
  - memória dividida em partes de tamanho fixo
  - memória lógica dividida em páginas
  - permite que o programa seja separado em áreas não contiguas
- **Segmentação**
  - programa é dividido em segmentos de tamanhos variados, cada um com o seu próprio espaço de endereçamento
- **Segmentação paginada**
  - Cada segmento é dividido fisicamente em páginas, o endereço é formado pelo número do segmento, número da página dentro desse segmento e o deslocamento dessa página

# Hardware envolvido na gerência de memória

- **MMU**
  - É um hardware
  - Unidade de Gerenciamento de memória
  - Mapeia endereços virtuais ou lógicos em endereços físicos

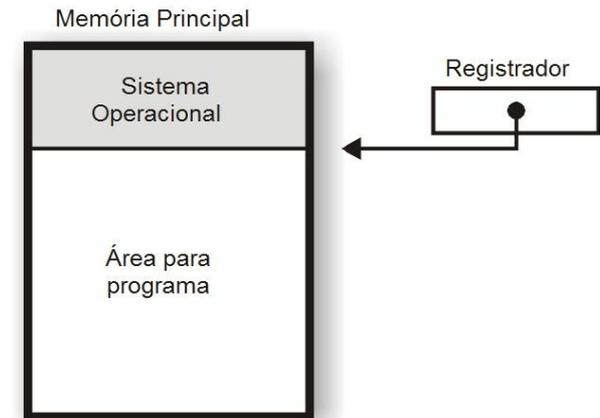


# MMU - Atuação



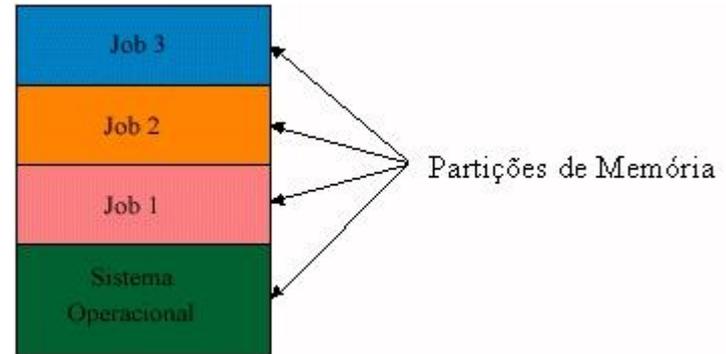
# Organização da memória - Monoprogramação

- Memória compartilhada entre o programa e S.O.
  - Toda memória livre como bloco único
  - Alocação Contígua simples
- Somente um programa é executado por vez, que utiliza toda a memória disponível (com exceção da parte reservada ao SO)
- Raro hoje em dia
  - Utilizada em sistemas embarcados
  - SOs monousuários e monotarefas
- Uso de overlay



# Organização da memória - Multiprogramação

- Vários processos (programas) compartilhando o espaço de memória
  - Enquanto um bloqueia (espera uma informação de E/S) , outro processo poderá utilizar a CPU
  - Aumenta a utilização da CPU



# Organização da memória - Multiprogramação

- **Com partições fixas/estáticas** (tamanhos diferentes)
  - Primeiros SOs multitarefa
  - Blocos de tamanhos pré-definidos / fixos
- Tipos
  - Absoluta
    - Endereço em relação ao endereço absoluto
    - Sempre na mesma partição
    - Em tempo de compilação
  - Relocável
    - Endereço em relação ao reg base
    - Programa pode rodar em qq partição que o caiba
    - Em tempo de execução
- Problema advindo: fragmentação interna
  - Perda de espaço dentro de uma área de tamanho fixo
  - Área alocada e utilizável



# Organização da memória - Multiprogramação

- **Partições variáveis / dinâmicas**
  - Memória principal é subdividida em segmentos de tamanho variado
  - Aloca os processos em blocos adjacentes
  - Memória dividida entre o SO e área livre
  - Divisão das partições em tempo de execução
  - Problema advindo: fragmentação externa
    - Começa a acontecer quando os programas forem terminando e deixando espaços cada vez menores na memória
    - Não permitindo o ingresso de novos programas
  - Fragmentação ocorre entre os segmentos (externa)
    - Área não alocada e não utilizável
    - Compactação de memória: movem os segmentos para próximo uns dos outros deixando espaço de memória contíguo, não-fragmentado disponível
    - requer dedicação exclusiva
      - São combinados em único espaço contíguo para endereços mais baixos
    - Fusão de lacunas adjacentes
      - Sobrecarga menor
    - Resolve também com os algoritmos de escolha de espaços vazios

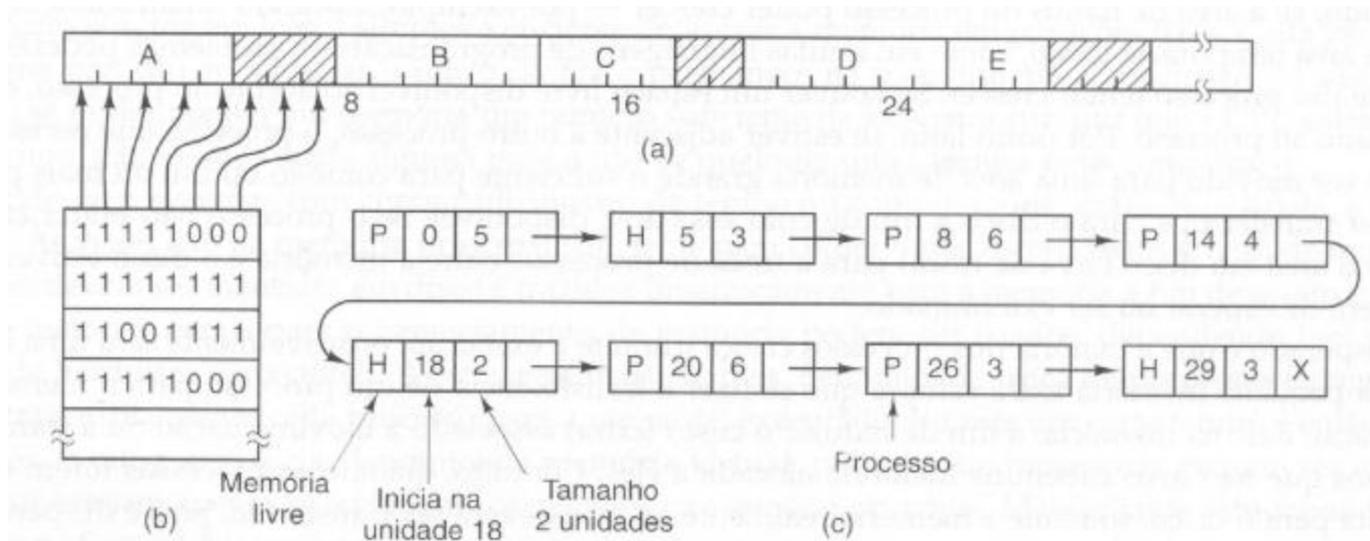
# Administração da memória – Controlando os espaços

- Gerenciamento com mapa de bits
  - Memória dividida em unidades de alocação, sendo cada unidade associada a um bit no mapa de bits
  - 0 ou 1 (Disponível/ Ocupado)
  - Procura pelo tamanho contíguo através de bits é um ponto negativo
  - Bom para endereçar, mas difícil de gerenciar
  - Problema: Operação muito lenta, principal argumento contra o mapa de bits
  - Tamanho da unidade de alocação
    - Grande
      - Mapa de bits menor
      - Maior desperdício de memória
    - Pequeno
      - Mapa de bits maior
      - Menor desperdício de memória
  - Como o mapa também fica na memória, então o tamanho da unidade de alocação é muito importante

# Administração da memória – Controlando os espaços

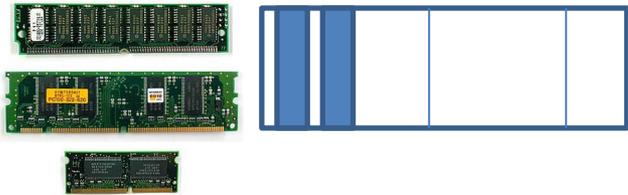
- Gerenciamento com lista encadeada
  - Gerenciamento de memória livre e desalocação é muito fácil
  - Mantém uma lista de segmentos de memória alocados e disponíveis
    - Lista simples aponta apenas para o próximo
    - Lista duplamente encadeada pode ser mais adequada
      - Torna mais fácil encontrar o item anterior da lista para concatenação
      - Apontam para a próxima lacuna e a anterior

# Comparativo do mapa de bits com a lista encadeada



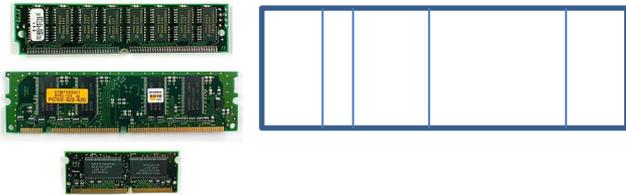
# Administração da memória – Algoritmos de escolha do espaço

- First fit (primeiro que couber)
  - Varre a tabela de espaços livres até encontrar a lacuna
  - Não existe encargo de ordenação dos espaços livres
  - Mais simples e rápido, pois pesquisa o mínimo possível
  - Pesquisa ao longo da lista um segmento livre que acomode o processo, ou seja, maior ou igual ao processo
  - A sobra transforma-se em outro segmento de memória livre



# Administração da memória – Algoritmos de escolha do espaço

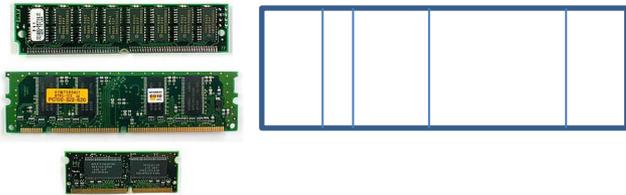
- Next fit (próximo que couber)



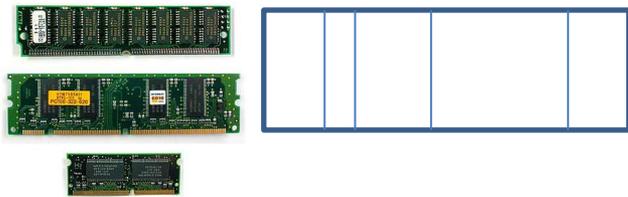
- Similar ao first fit
- Diferença: memoriza a posição do segmento encontrado
- Recomeça busca a partir desta posição
- Desempenho ligeiramente inferior ao First Fit

# Administração da memória – Algoritmos de escolha do espaço

- Best fit (melhor que couber)
  - Varre toda a tabela de espaços livres
  - Percorre a lista inteira e escolhe o menor segmento livre de memória possível
  - Mais lento que o first fit
  - Maior desperdício de memória que o first e next fit
  - Gera minúsculos segmentos

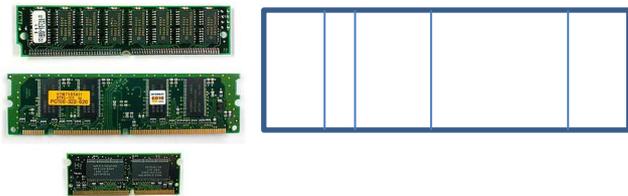


# Administração da memória – Algoritmos de escolha do espaço



- Worst fit (pior que couber)
  - Varre toda a tabela de espaços livres
  - Escolhe maior segmento disponível, de maneira que quando for alocado, gere um segmento suficientemente grande para alocar outro processo

# Administração da memória – Algoritmos de escolha do espaço



- Quick fit (o que mais rápido couber)

- Mantém listas separadas para alguns tamanhos de segmentos disponíveis
- Em geral os tamanhos mais solicitados



# Bateria de questões de aprendizagem 1

Sistemas Operacionais

## INMETRO – CESPE 2010 – Pesq TMQ – Ciência da Computação

1. Considere um sistema com swapping, no qual as seguintes partições vazias de tamanho fixo estão na memória, na ordem apresentada: 20K, 14K, 35K, 8K, 17K, 39K, 22K e 27K. Se um processo solicitar a alocação de uma área de memória de 21K, o algoritmo de alocação de memória que faz a alocação minimizando a fragmentação interna é
  - A. next-fit.
  - B. first-fit.
  - C. worst-fit.
  - D. best-fit.
  - E. last-fit.

**CASA DA MOEDA – CESGRANRIO 2009 – Analista – Banco de Dados**

2. Determinado sistema operacional apresenta a seguinte lista de segmentos de memória disponíveis, em sequência:

10 KB, 20 KB, 40 KB, 80 KB, 160 KB

Nesse momento, um processo solicita alocação de memória para 16 KB.  
Considerando-se o uso do algoritmo first fit, que segmento será utilizado na alocação?

- A. 10 KB
- B. 20 KB
- C. 40 KB
- D. 80 KB
- E. 160 KB

## SESAU – RO – FUNCAB 2009 – Analista de Sistemas

3. No gerenciamento de memória com listas encadeadas, o algoritmo que inicia a procura a partir do local onde foi feita a última locação e escolhe o bloco que é grande o suficiente para o processo a ser alocado é chamado de:
- A. First Fit;
  - B. Next Fit;
  - C. Best Fit;
  - D. Quick Fit;
  - E. Worst Fit.

## BNDES – CESGRANRIO 2008 – Analista de Sistemas - Desenvolvimento

4. Um sistema operacional que usa segmentação de memória tem a seguinte configuração:

segmento 1 : 100 KB / segmento 2 : 50 KB / segmento 3 : 200 KB / segmento 4 : 50 KB

A memória é preenchida com segmentos de diversos tamanhos, colocados na ordem abaixo.

20 KB, 50 KB, 10 KB, 50 KB, 20 KB, 20 KB, 40 KB

Qual o espaço livre de cada segmento, respectivamente, para as estratégias abaixo?

I: best-fit

II: first-fit

III: worst-fit

- A. I: 1=30 KB, 2=0 KB, 3=160 KB, 4=0 KB II: 1=0 KB, 2=0 KB, 3=140 KB, 4=50 KB III: 1=60 KB, 2=50 KB, 3=30 KB, 4=50 KB
- B. I: 1=0 KB, 2=0 KB, 3=160 KB, 4=30 KB II: 1=0 KB, 2=0 KB, 3=140 KB, 4=50 KB III: 1=60 KB, 2=50 KB, 3=30 KB, 4=50 KB
- C. I: 1=30 KB, 2=0 KB, 3=160 KB, 4=0 KB II: 1=60 KB, 2=50 KB, 3=30 KB, 4=50 KB III: 1=0 KB, 2=0 KB, 3=140 KB, 4=50 KB
- D. I: 1=0 KB, 2=0 KB, 3=160 KB, 4=30 KB II: 1=0 KB, 2=0 KB, 3=140 KB, 4=50 KB III: 1=20 KB, 2=0 KB, 3=120 KB, 4=50 KB
- E. I: 1=20 KB, 2=0 KB, 3=120 KB, 4=50 KB II: 1=60 KB, 2=50 KB, 3=30 KB, 4=50 KB III: 1=0 KB, 2=0 KB, 3=140 KB, 4=50 KB

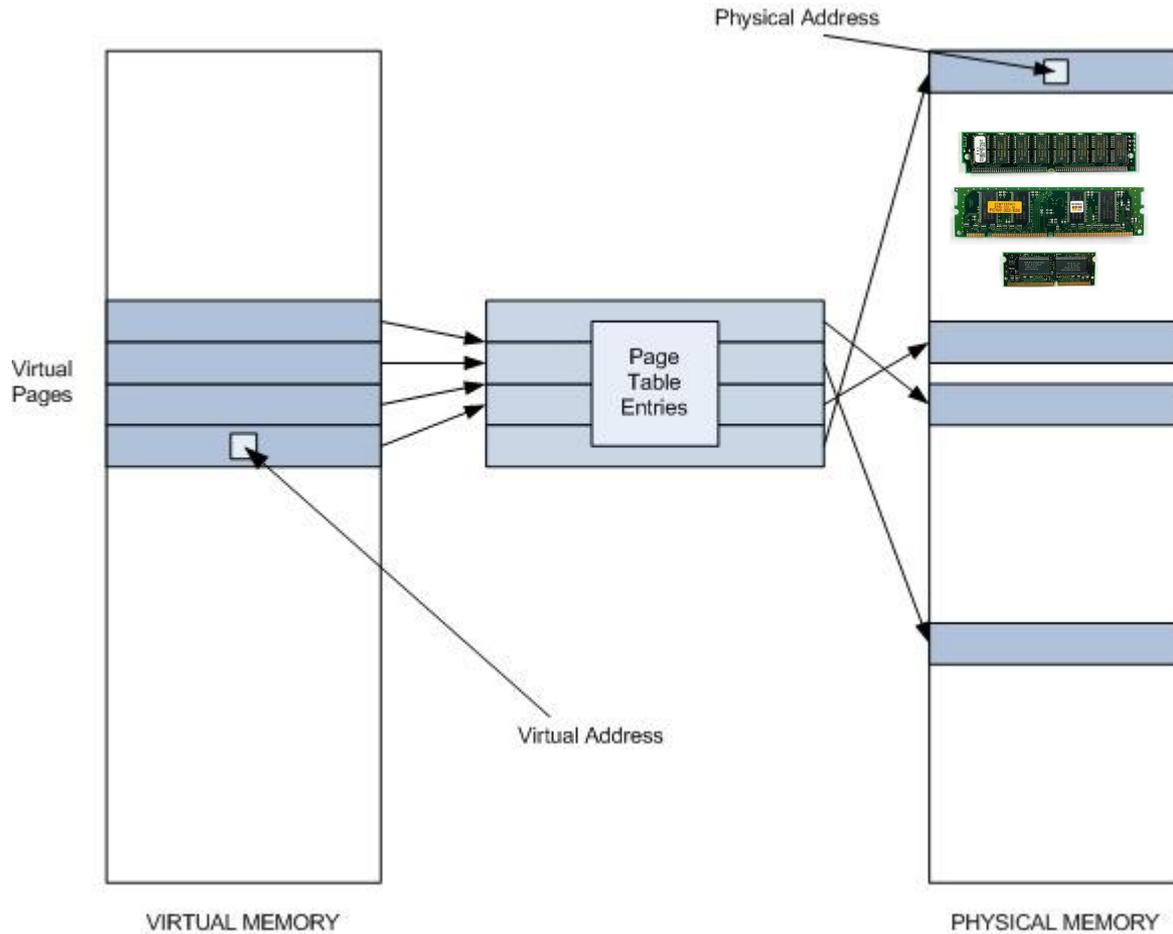
# Gabarito

1. D
2. B
3. B
4. A

# Endereçamento Virtual

- Espaço de endereçamento virtual
  - Programas geram endereços virtuais
  - Constituído pelos endereços virtuais que os processos podem endereçar
  - É dividido em páginas (512 bytes a 64 KB)
  - As unidades correspondentes em memória física são denominadas molduras de página
  - Páginas e molduras sempre do mesmo tamanho
- **Não confundir memória virtual (estratégia) x endereçamento virtual x memória virtual (extensão em disco)**

# Endereçamento Virtual



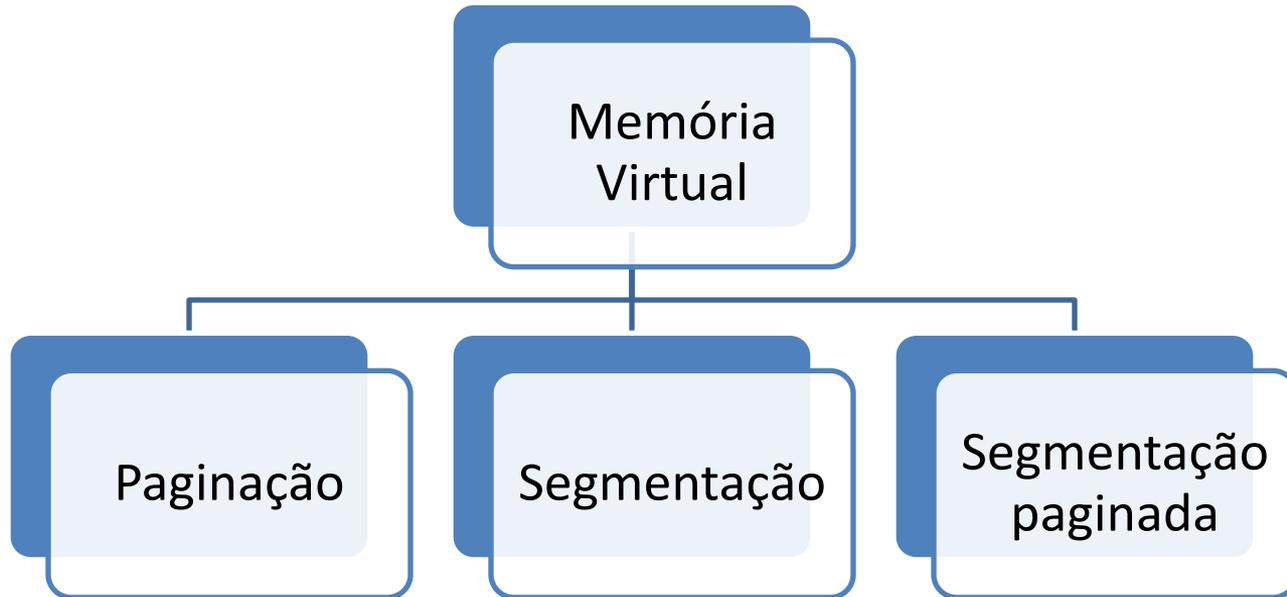
# Endereçamento Virtual

- Permite programas (código + dados + pilha) maiores que a memória disponível
  - RAM + Disco
  - Ilusão de memória única
  - SO 32 bits impõe ao aplicativo 4 GB
  - Não existe tamanho, apenas capacidade de endereçamento
- Solução em Hardware: Gerenciado pela MMU
- Solução em software: Sistema Operacional toma conta

# Swapping

- Swapping (troca MD)
  - No windows: É um arquivo de uso exclusivo pelo SO
  - No Linux: Partição separada ou arquivo específico para esse fim
  - Não é possível ter nenhum outro tipo de arquivo nesse local
  - Thrashing = muitos page faults e alta taxa de paginação
  - Swap out x swap in

# Memória Virtual



# Paginação - Considerações

- Espaço real e virtual são divididos em blocos de MESMO tamanho
- Unidades de endereçamento virtual = Páginas
- Unidades de memória física = Molduras de Página
- MMU - responsável pelo mapeamento - tabela de páginas
- Cada processo tem sua tabela de páginas
- Tab Página: Bit presente/ausente / No. Moldura / Modificada / Referenciada
- Página virtual não mapeada = page fault
- Página modificada = bit modificado = 01 = sujo = atualizar disco
- Tamanho da página pode ser escolhido pelo SO

# Paginação - Considerações

- Memória associativa ou TLB
  - Cache que armazena uma parte da tabela de páginas
  - Dentro do MMU
  - Dispositivo em hardware para mapear os endereços virtuais em físicos, sem passar pela tabela de páginas
  - Buffer de tradução de endereços, geralmente localizada dentro da MMU, consiste em um conjunto de registradores, ~64 entradas
  - Leitura das entradas se dá de forma paralela
  - Entradas possuem índices
  - Na ausência de página na TLB, a MMU busca na tabela de páginas.

# Envolvimento do SO na paginação

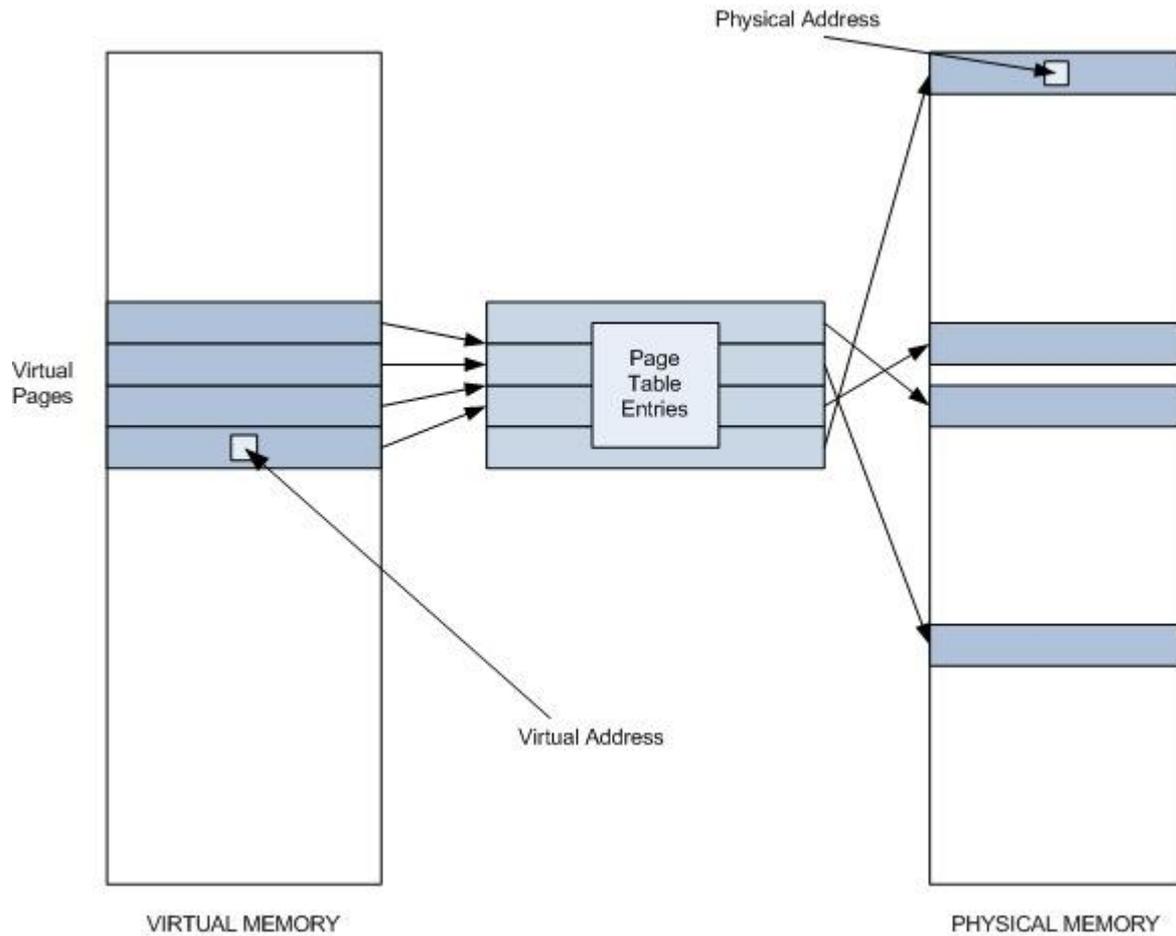
- Criação do processo
- Execução do processo
- Ocorrência de page fault
- Finalização do processo



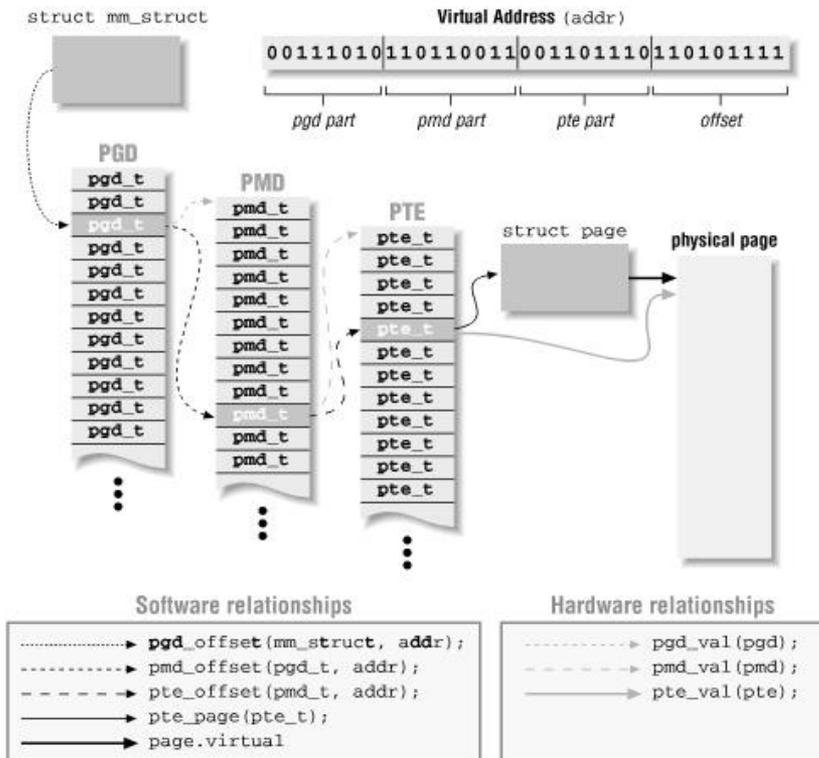
# A Tabela de Páginas

- Objetivo: Mapear páginas virtuais em molduras de páginas
- Endereço virtual
  - Constituído pelo número da página (bits mais significativos) e um deslocamento (bits menos significativos)
  - O deslocamento é o ponto que será lido (NNNNNNNNDDDDDD)
  - Como encontrar o endereço físico
    - 1. O número da página virtual é utilizado como índice dentro da tabela de páginas a fim de obter a moldura de página
    - 2. O número da moldura de página é concatenado aos bits do deslocamento. Endereço físico que será enviado à memória





# A Tabela de Páginas



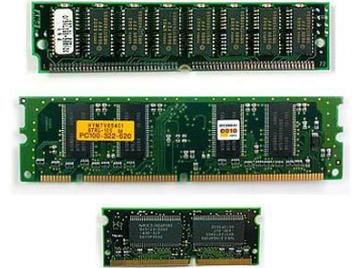
- Tabela de páginas multinível
  - Minimiza o problema de armazenar tabelas de páginas muito grandes na memória
  - Tabelas de páginas muito grandes apresentam mapeamento mais lento
  - Evita que todas as tabelas sejam mantidas na memória
  - Mapeamento deve ser rápido
  - Apenas partes das tabelas na RAM
  - Tabela de diretórios na RAM
- Solução: Tabelas de páginas multiníveis

# A Tabela de Páginas

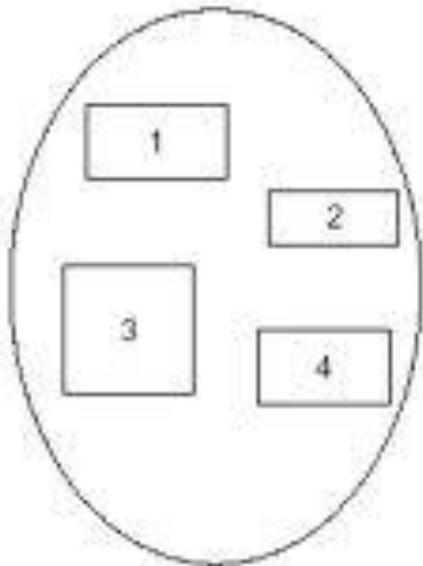
- Estrutura básica de uma entrada na tabela
  - Número da moldura de página (mais importante)
  - Bit presente/ausente
  - Bits de proteção
  - Bit cache desabilitado
  - Bit Modificada (bit sujo)
  - Bit Referenciada
- Observações
  - Cada processo possui sua própria tabela de páginas e existe uma tabela geral
  - Deve estar na memória RAM

# Alguns inconvenientes da paginação

- Fragmentação interna (o processo não utiliza a página toda)
  - Maior ou menor fragmentação é consequência do tamanho de páginas
    - Páginas pequenas = grandes tabelas
    - Páginas grandes = desperdício
- Thrashing
  - Excessiva transferência de página entre a memória principal e secundária
  - Ocorre quando existem mais processos competindo por memória do que o espaço disponível
  - Ocorre na paginação ou segmentação
  - Solução: aumentar a memória



# Segmentação



Espaço do Usuário



Espaço de Memória Física

- Sequência linear de endereços, de 0 a algum máximo
- Podem crescer ou reduzir independentemente
- Facilita o compartilhamento de procedimentos ou dados entre vários processos.
- Fragmentação externa (checkerboarding)
  - Memória dividida em regiões com segmentos e regiões com lacunas
  - Desperdiça memória nas lacunas
  - Solução: compactação

# Segmentação

- Blocos de tamanho variável
  - Tamanho varia em função do bloco da aplicação
  - O que pode ser maior que o EE é a soma dos espaços de endereçamento de todos os processos
  - O espaço de endereçamento de cada processo tem que ser menor ou igual ao limite da arquitetura
- Mapeamentos
  - Tabela de mapeamento de segmentos
  - Endereços são compostos por NSV (número do segmento virtual) e um deslocamento
  - Cada segmento possui uma ETS (entrada na tabela de segmentos)

# Segmentação Paginada

- Memória dividida em Segmentos
- Segmentos divididos em páginas
- Segmentos devem ser múltiplos da unidade de alocação (4KB...8KB...etc)
- Simplicidade da paginação + controle de acesso da segmentação

# Bateria de questões de aprendizagem 2

Sistemas Operacionais

## PETROBRÁS– CESGRANRIO 2012 – Analista de Sistemas Júnior - Infra

1. Quando os programas esperam por memória livre para serem executados devido à insuficiência de memória principal, o sistema operacional pode solucionar esse problema com a aplicação da técnica de
  - A. falha de página
  - B. segmentação
  - C. swapping
  - D. thrashing
  - E. partição

## **BNDES– CESGRANRIO 2011 – Profissional Básico – Análise e Desenvolvimento de Sistemas**

2. Na memória virtual por paginação, o espaço de endereçamento virtual e o espaço de endereçamento real são divididos em blocos do mesmo tamanho chamados páginas. Na memória virtual por segmentação, o espaço de endereçamento é dividido em blocos de tamanhos diferentes chamados segmentos. Na memória virtual por segmentação com paginação, o espaço de endereçamento é dividido em
- A. segmentos e, por sua vez, cada segmento dividido em páginas, o que elimina o problema da fragmentação externa encontrado na segmentação pura.
  - B. segmentos e, por sua vez, cada segmento dividido em páginas, o que elimina o problema da fragmentação interna encontrado na segmentação pura.
  - C. segmentos e, por sua vez, cada segmento dividido em páginas, o que elimina o problema da fragmentação interna encontrado na paginação pura.
  - D. páginas e, por sua vez, cada página dividida em segmentos, o que elimina o problema da fragmentação externa encontrado na segmentação pura.
  - E. páginas e, por sua vez, cada página dividida em segmentos, o que elimina o problema da fragmentação interna encontrado na segmentação pura.

# FINEP – CESGRANRIO 2011 – Analista - Suporte

3. Memória virtual é uma técnica de gerência de memória que mantém apenas parte do código de execução e parte da área de dados dos programas em memória real. Nesse contexto, a técnica de paginação divide o espaço de endereçamento
- A. virtual em blocos com tamanho igual ao tamanho dos blocos do espaço de endereçamento real
  - B. virtual em blocos com tamanho igual ao tamanho do espaço de endereçamento real
  - C. virtual em blocos com tamanho igual à metade do tamanho do espaço de endereçamento real
  - D. real em blocos com tamanho igual ao tamanho do espaço de endereçamento virtual
  - E. real em blocos com tamanho igual à metade do tamanho do espaço de endereçamento virtual

## TRT 14 – FCC 2011 – Técnico Judiciário - TI

4. A Memory Management Unit (MMU) tem como função
- A. mapear páginas virtuais em molduras de página.
  - B. mapear os endereços virtuais para endereços físicos de memória.
  - C. dividir a memória em partições de tamanhos variados.
  - D. pesquisar e selecionar o job que melhor se ajuste ao tamanho da partição.
  - E. gerenciar os registradores de base e registradores de limite.

## TRT 14 – FCC 2011 – Técnico Judiciário - TI

5 - Quando um job chega para ocupar uma partição de memória, ou ele é colocado em uma fila de entrada da menor partição capaz de armazená-lo ou ele é colocado em uma fila de entrada única para todas as partições. No contexto de gerenciamento de memória trata-se de uma afirmativa típica da

- A. Multiprogramação com partições fixas.
- B. Monoprogramação sem troca.
- C. Multiprogramação com troca.
- D. Monoprogramação sem paginação.
- E. Multiprogramação com partições dinâmicas.

## PGE DF – IADES 2011 – Analista Jurídico – Análise de Sistemas

6. A maioria dos sistemas operacionais da atualidade utiliza o recurso chamado Memória Virtual. Uma das funções da Memória Virtual é a paginação ou troca (swapping). Assinale a alternativa que contém a afirmação correta a respeito do swapping.
- A. Swapping possibilita ao sistema operacional e às aplicações o uso de mais memória do que a fisicamente existente em um computador.
  - B. A principal função do swapping é impedir que um processo utilize endereço de memória que não lhe pertença.
  - C. O swapping é uma técnica de endereçamento que faz com que cada processo enxergue sua área de memória como um segmento contíguo.
  - D. Swapping é a capacidade de troca de componentes de hardware de um computador, mesmo que o mesmo esteja ligado.
  - E. O swapping é um mecanismo necessário em computadores de 64 bits que permite o endereçamento de memórias superiores a 4 Gbytes.

## INMETRO – CESPE 2010 – Pesq TMQ – Infra e redes de TI

7. Assinale a opção correta com relação aos conceitos de gerenciamento de memória.
- A. Os sistemas de gerenciamento de memória podem ser divididos em quatro grupos: paginação, swapping, tempo real e manipuladores de interrupção.
  - B. As funções do gerenciador de memória incluem determinar quais partes da memória estão ou não em uso; alocar memória para os processos quando dela necessitem; desalocar memória quando processos forem encerrados ou deixem de usá-la ou quando terminem. Ele pode, ainda, controlar as trocas de dados entre a memória principal e o disco rígido, quando a memória principal não é grande o suficiente para conter todos os processos.
  - C. Tipicamente, em um ciclo de execução do processador, primeiramente uma instrução será carregada do processador para a memória; a seguir, a instrução será decodificada e poderá fazer que operandos sejam carregados do processador; finalmente, após a execução da instrução sobre os operandos, resultados eventualmente poderão ser armazenados de volta na memória.
  - D. O carregamento estático pode ser empregado para se alcançar melhor utilização do espaço em memória, por meio do carregamento, na memória, de uma rotina que aguarda ser chamada. Nesse caso, todas as rotinas são mantidas em memória em um formato relocável.
  - E. Quando a memória física não for suficiente para conter todo o programa a ser executado, pode-se resolver esse problema com o uso de gerenciamento de tabelas.

## TRT 9 – FCC 2010 – Analista Judiciário – Tecnologia da Informação

8. No contexto de gerenciamento de memória, é correto afirmar:

- A. Cada entrada em uma tabela de segmentos possui a "base", que contém o endereço físico inicial do segmento residente na memória e o "limite", que especifica o tamanho do segmento.
- B. O swapping é uma técnica utilizada para mudar a localização dos processos na memória, agrupando-os em um único segmento e, assim, otimizar a execução dos processos concorrentes.
- C. O hardware MMU ( Unidade de Gerência de Memória ) tem como função mapear os endereços físicos em endereços virtuais para serem vistos pela memória.
- D. Na realocação dinâmica, todas as rotinas são carregadas na memória principal e aquelas que não são usadas são agrupadas em segmentos contíguos da memória.
- E. A alocação contígua à memória principal é dividida em duas partes: a parte alta para o sistema operacional e o vetor de interrupções, e parte baixa para os processos do usuário.

# Gabarito

1. C

2. A

3. A

4. B

5. A

6. A

7. B

8. A

# Escalonamento

- Útil, quando 2 ou + processos estão em estado de pronto
- Parte do S.O. = Escalonador
  - Escolha do processo
  - Uso eficiente da CPU
- Processos orientados a CPU x E/S
- Quando escalonar?
  - Quando se cria um novo processo
  - No término de um processo
  - Quando um processo bloqueia para E/S
  - Quando ocorre uma interrupção de E/S (liberação do processo bloqueado)



# Escalonamento

- Multiprogramação
  - CPU não é um recurso escasso
- Objetivos Gerais
  - Justiça
  - Aplicação da política
  - Equilíbrio - Manter partes do sistema ocupadas
- Prioridades
  - Estáticas: Prioridade fixa
  - Dinâmicas: Prioridade pode ir mudando
  - Inversão de prioridade
    - Aumento de prioridade de um processo menos importante
    - "se livrar dos chatos"



# Entendendo os escalonamentos



# Escalonadores - Swapper

- Escalonador de memória / intermediário / médio prazo
- Ram <--> Disco (memória virtual em disco)
- Seleciona entre os processos que estão na memória virtual, reduz o grau de multiprogramação
- Determina quais processos ficam total ou parcialmente na memória durante a sua execução
- Está intimamente ligado à gerência de memória



# Escalonadores - Scheduler

- RAM <--> fila de pronto
- define o grau de multiprogramação
- Executado quando um processo é criado
- Determina quando um processo novo é efetivamente admitido no sistema para processamento e disputa de recursos
- Quanto maior o número de processos admitidos menor é a porcentagem de tempo de CPU para cada um deles



# Escalonadores – Dispatcher

- Escalonador de CPU / baixo / curto prazo
- Pronto <--> CPU
- O escalonador de curto prazo faz decisões de escalonamento muito mais frequentemente que os de médio e longo prazo
- È invocado na ocorrência de eventos importantes
  - Interrupções do relógio (fim da fatia de tempo)
  - Chamadas ao Sistema (Ex: operação de E/S)
  - Interrupções de I/O (Término de operação de E/S)



# A memória secundária

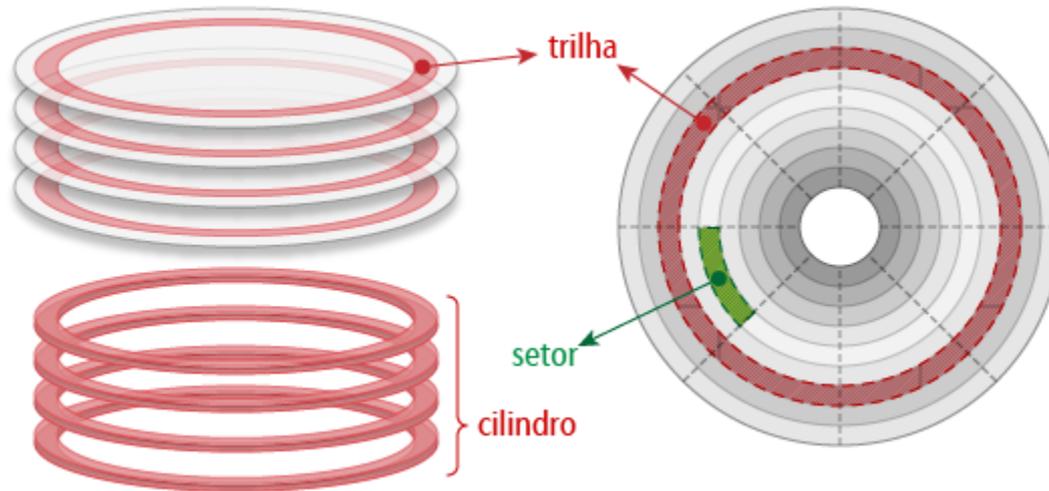
- É toda memória que não pode ser acessada diretamente pelo processador: HDs, pen drives, disquetes, etc...
- O HD, por contemplar peças mecânicas e características mais sofisticadas de desempenho será o nosso representante das mídias secundárias (das bancas também).
- COMPONENTES do Tempo de acesso =  $T_{busca} + T_{latência} + T_{transmissão}$   
(Influência direta no desempenho)



# Estrutura do HD



# Estrutura do HD



# Componentes do desempenho do HD

- **Seek Time** - tempo necessário para colocar a cabeça de leitura no cilindro desejado
- **Rotational Delay** - posicionada a cabeça, será necessário localizar o setor pretendido
  - Influenciado pelas RPM do disco
  - Aumento da velocidade linear
  - Manutenção da velocidade angular
- **Transfer Time** - tempo de leitura de um setor



# Técnicas de redução de tempo de acesso a disco

- Período decorrido entre a ordem de acesso e o final da transferência dos dados
  - **Cache de blocos/buffer:** Área intermediária entre periféricos
    - **Write-through:** Escrita direta para o cache da controladora e disco
    - **Write-back:** SO não grava imediatamente no disco, mas acumula dados no cache da controladora para PERIODICAMENTE lançar no disco
  - **Leitura antecipada de blocos**
    - Só funciona para arquivos sequenciais
    - Para arquivos com acesso aleatório a situação se agrava mais ainda
  - **Redução do movimento do braço de disco**



# Outras técnicas auxiliares para melhoria de desempenho dos HDs

- Uso de tecnologias de maior desempenho: SATA II, III, SCSI, SSD
- Aumentar a densidade dos meios de armazenamento
- RAID por software ou hardware
- Uso de técnica de antecipação do braço de disco
- Desfragmentação de disco
- Trabalho com dados compactados (diferente de comprimidos)
- Usar blocagem para agrupar segmentos de dados em setores contíguos (cilindros)
- Adicionar novos discos
- Transferir serviços para um outro servidor



# Bateria de questões de aprendizagem 3

Sistemas Operacionais

## Pref Ibiporã – AOCF 2011 – Analista de Sistema

1. Sobre Gerência do Processador em sistemas operacionais, analise as assertivas e assinale a alternativa que aponta as corretas.
  - I. Um algoritmo de escalonamento tem como principal função decidir qual dos processos prontos para execução deve ser alocado à UCP.
  - II. Em sistemas multiprogramáveis não existem algoritmos de escalonamento de processos.
  - III. Na maioria dos sistemas é desejável que o processador permaneça a maior parte do seu tempo ocupado.
  - IV. Cada sistema operacional necessita de um algoritmo de escalonamento adequado ao seu tipo de processamento.
- A. Apenas I e II.
- B. Apenas II e III.
- C. Apenas I, II e IV.
- D. Apenas I, III e IV.
- E. I, II, III e IV.

## FINEP – CESGRANRIO 2011 – Analista de Suporte

2 Considere as afirmações abaixo sobre os critérios que devem ser adotados em uma política de escalonamento de processos (tarefas) em sistemas multiprogramáveis.

- I - A política de escalonamento de processos tem como critério maximizar a utilização do processador, mantendo-o ocupado a maior parte do tempo e balanceando sua utilização entre os diversos processos.
- II - A política de escalonamento de processos busca maximizar o número de processos (tarefas) executados em um determinado intervalo de tempo.
- III - Maximizar o tempo que um processo leva desde sua admissão até o seu término é um critério típico da política de escalonamento de processos.
- IV - Minimizar o tempo de resposta, oferecendo tempos de resposta razoáveis para os usuários é um dos critérios da política de escalonamento de processos.

Estão corretas as afirmações

- A. I e II, apenas.
- B. I e III, apenas.
- C. II, III e IV, apenas.
- D. I, II e IV, apenas.
- E. I, II, III e IV.

## INMETRO – CESPE 2010 – Pesq TMQ – Ciência da Computação

3. Diferentes algoritmos de escalonamento de CPU possuem diferentes propriedades e a escolha de um determinado algoritmo pode favorecer uma classe dos processos em detrimento de outra. Assinale a opção que apresenta um algoritmo capaz de produzir starvation.

- A. prioridade
- B. Round-Robin
- C. primeiro a chegar, primeiro a ser atendido
- D. eleição
- E. múltiplas filas com realimentação

## TRT 22 – FCC 2010 – Analista Judiciário – TI

4. O sistema operacional é responsável por uma ou mais das seguintes atividades relacionadas ao gerenciamento de disco:

(I) Gerenciamento do espaço livre.

(II) Alocação do armazenamento.

(III) Interpretação de comandos.

(IV) Escalonamento do disco.

Está correto o que se afirma em

- A. I, II e III, apenas.
- B. I, III e IV, apenas.
- C. I, II e IV, apenas.
- D. II, III e IV, apenas.
- E. I, II, III e IV.

## INMETRO – CESPE 2010 – Pesquisador – Metrologia em Informática

- 5 Muitos processos podem gerar requisições para ler e escrever dados em um disco simultaneamente. A política de escalonamento dos acessos a discos rígidos tem um impacto importante no throughput de um sistema. A respeito de estrutura de discos e escalonamento de discos, assinale a opção correta.
- A. Tempo de busca é o tempo necessário para se deslocar o dado de sua posição atual até posicioná-lo no local da cabeça de leitura e gravação.
  - B. Atraso ou latência rotacional é o intervalo de tempo necessário para alinhar o início do setor desejado com o cabeçote de leitura/escrita.
  - ~~C. Em um disco de 41 trilhas, se o cabeçote de leitura/gravação se encontra na trilha 15 e as requisições de acesso às trilhas 4, 7, 11, 14, 35 e 40 estão aguardando seu processamento, o uso do escalonamento LOOK fará que a ordem de acesso seja 14, 11, 7, 4, 35 e 40, pois, nesse tipo de escalonamento, os acessos a disco são ordenados conforme sua distância relativa.~~
  - ~~D. Se o cabeçote de leitura/gravação se encontra na trilha 15 e as requisições de acesso às trilhas 4, 7, 11, 12, 25 e 39 estão aguardando seu processamento, o uso do escalonamento shortest seek time first (SSTF) fará que a ordem de acesso seja 25, 39, 12, 11, 7 e 4.~~
  - ~~E. O algoritmo N-step SCAN incorpora as requisições no percurso do braço à medida que ele se desloca, não segurando as requisições até que o braço inverta a direção de seu caminho.~~

# Gabarito

1. D
2. D
3. A
4. C
5. B