

Exercícios - Cap I

- 1.1, 1.2, 1.3 (somente letras (a), (b) e (c))
- 1.5 1.7, 1.8 e 1.12

Sistemas Operacionais

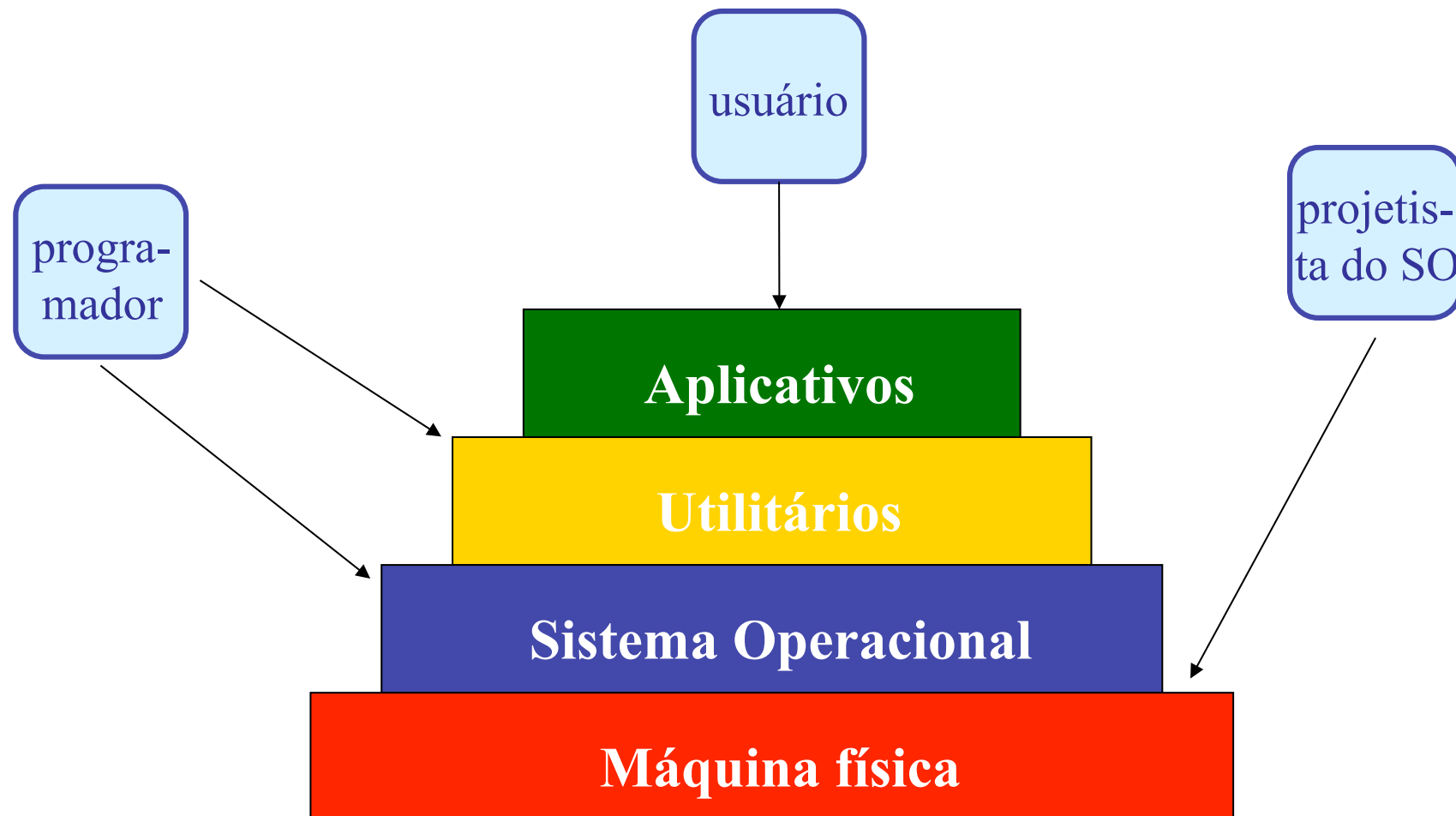
Visão geral e evolução dos SOs

Sistema Operacional?

- Um programa que controla a execução dos programas de aplicação
- Uma interface entre o usuário e o *h/w*
- Um programa que mascara os detalhes do *h/w*

Duas visões: gerenciador de recursos e máquina virtual

SO como máquina virtual



Máquina virtual: serviços

- Criação de programas
- Execução de programas
- Acesso a dispositivos de E/S
- Acesso controlado a arquivos
- Acesso ao sistema
- Detecção e correção de erros
- Contabilidade

Máquina virtual: serviços

- Criação de programas
 - SO oferece facilidades: editores e depuradores
 - tipicamente estes serviços não são parte do SO e sim dos **utilitários**
 - contudo, são **acessíveis** através do SO

Máquina virtual: serviços

- Execução de programas
 - carregamento do programa em memória
 - arquivos e dispositivos de E/S devem ser iniciados
 - outros recursos devem ser preparados
 - SO **gerencia** estas ações para o usuário

Máquina virtual: serviços

- Acesso a dispositivos de E/S
 - cada dispositivo tem seu próprio conjunto de instruções ou sinais de controle
 - SO **esconde** estas ações e usuário só executa leituras e escritas

Máquina virtual: serviços

- Acesso controlado a arquivos
 - usuário não se preocupa com a **natureza** do dispositivo de E/S (disco, fita, ...)
 - usuário não se preocupa com **formato** do arquivo no dispositivo
 - mecanismos de **proteção** em caso de múltiplos usuários

Máquina virtual: serviços

- Acesso ao sistema (recursos)
 - SO **controla** acesso ao sistema como um todo e a recursos específicos em particular
 - **proteção** contra acesso não autorizado
 - **resolução** de conflitos em caso de disputa

Máquina virtual: serviços

- Detecção e correção de erros
 - erros de *h/w*: memória, dispositivos, ...
 - erros de *s/w*: estouro aritmético, acesso proibido a certas posições de memória
 - **correção** da situação com mínimo de impacto no sistema

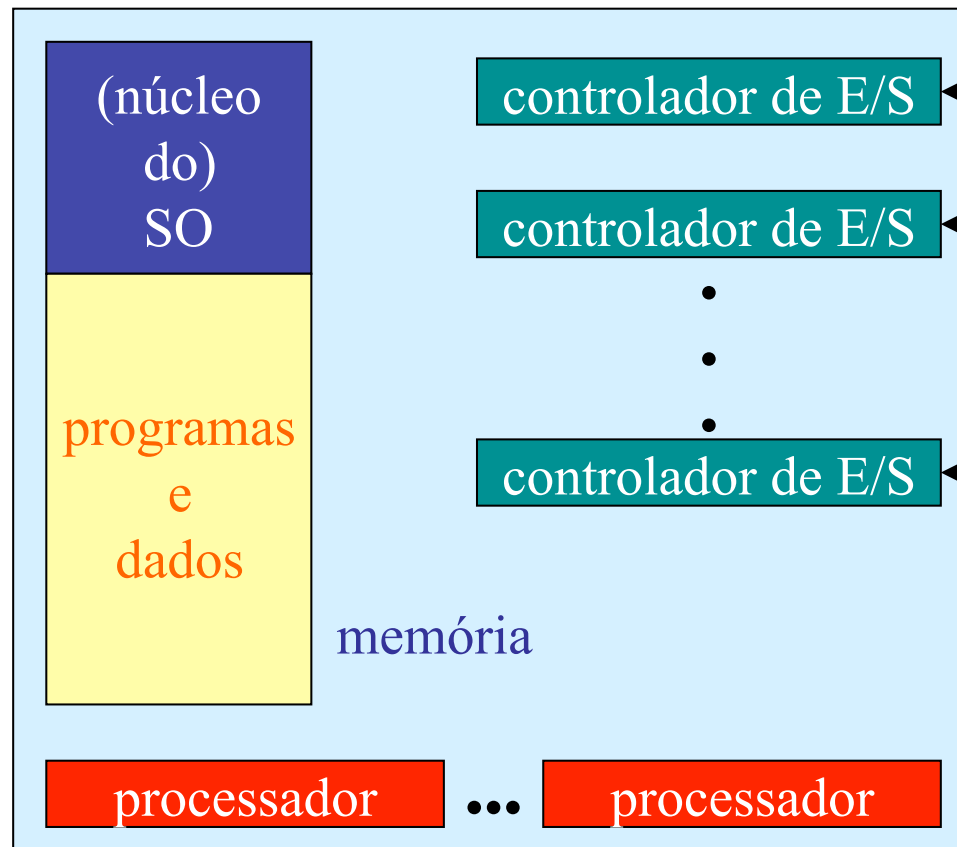
Máquina virtual: serviços

- Contabilidade
 - coleta de estatísticas
 - monitoramento de desempenho
 - uso: melhoria de desempenho, melhorias futuras
 - tarifação em um sistema multiusuário

SO: gerenciador de recursos

Sistema computacional

Dispositivos de E/S



Núcleo

- funções mais utilizadas
- residente na MP
- decide utilização de recursos e escalonamento

Sistema operacional

- É um programa!
- Direciona o processador no uso dos recursos do sistema e sobre o momento de executar outros programas
- SO libera o processador para que outros programas possam executar

Evolução de um SO

- Um SO deve evoluir ao longo do tempo para (novas versões):
 - receber novos tipos de *hardware* (e.g., novo terminal gráfico)
 - atender novos serviços (e.g., sistema de janelas)
 - reparar defeitos

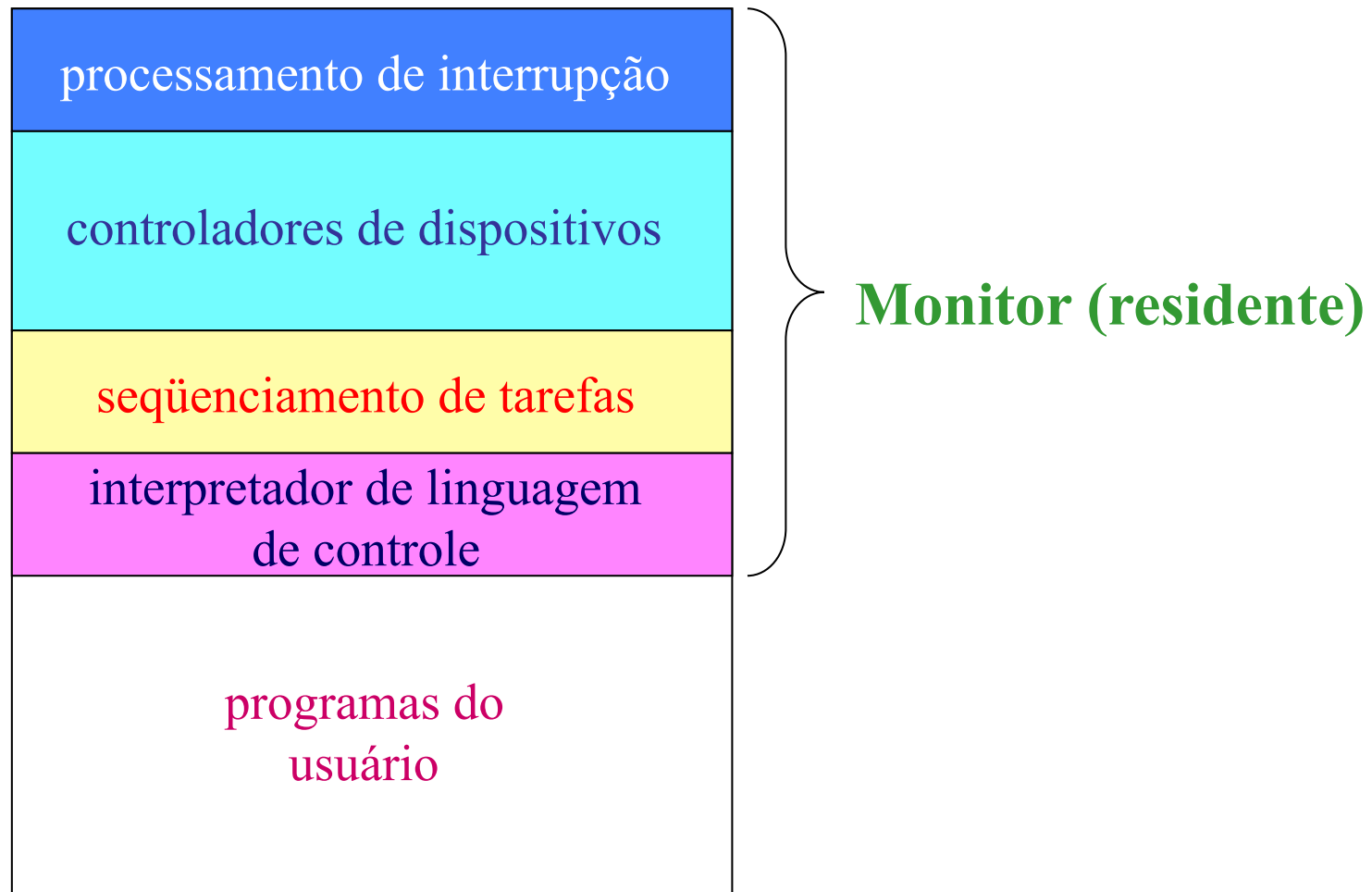
Inicialmente:

- Usuário fazia tudo - processamento serial!
- Ociosidade da máquina

Monitores

- *Software* que controla a execução de outros programas
- SO de lote (*batch*): *jobs* (tarefas) são carregados juntos
- Monitor é residente em memória principal
- Utilitários são carregados à medida da necessidade
- Usuário submete seu *job* e provê entrada
- Grau de ociosidade menor, mas execução seqüencial dos diferentes *jobs*

Monitor: mapa de memória



JCL: job control language

- Tipo especial de linguagem de programação
- Direciona o monitor:
 - que compilador usar
 - que dados usar
 - que dispositivos montar

Características de h/w desejáveis

- Proteção de memória
 - não permitir que a área ocupada pelo monitor seja alterada
- Temporização (início de multiprogramação)
 - prevenir um *job* de monopolizar o sistema
 - ocorrência de interrupção quando o tempo termina

Características de h/w desejáveis

- Instruções privilegiadas
 - executadas somente pelo monitor
 - e.g., instruções de E/S
 - ocorrência de interrupção caso o programa do usuário tente uma dessas instruções
- Interrupções
 - flexibilidade para controlar programas do usuário

Monitores

Vantagens:

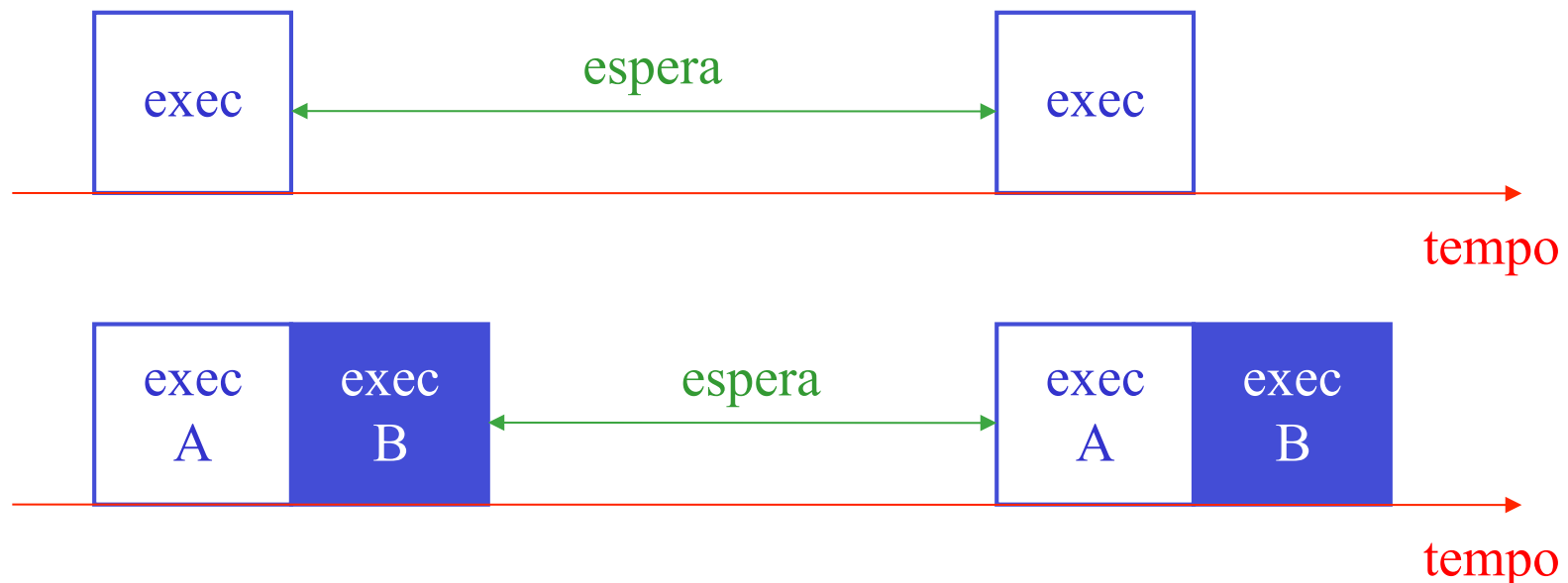
- Proteção
- Independência
- Organização

Desvantagens

- Monitor na Memória
- Sequenciamento de processos
- Ainda: sobrecarga de troca entre monitor e processos

Multiprogramação

- Permite que o processador execute outro programa enquanto um espera por E/S



Contudo ...

- Necessidade de *hardware* extra, como:
 - E/S por interrupção
 - gerenciamento de memória
- Necessidade de *software* extra, como:
 - escalonamento de processos
 - proteção de arquivos
 - sincronização entre processos

Time-sharing

- Uso de multiprogramação para atendimento de tarefas interativas
- UCP é compartilhada
- Acesso via terminais
- SO deve atender a um objetivo: minimizar o tempo de espera de cada usuário
 - **Time slice** para cada usuário → vantajoso pois usuários são "lentos"

Funções Principais em um SO

- Processos
- Concorrência
- Escalonamento de Processos
- Gerenciamento de Memória
- Memória Virtual
- Segurança e Proteção

Processos

- Mais geral que programa
- Consiste em um código executável e seus dados associados, além de um contexto de execução
- Linhas principais de desenvolvimento de sistemas de computadores que levaram a especificação de processos:
 - Multiprogramação em batch: maximizar utilização devido E/S
 - Time sharing: utilização assíncrona do sistema por usuários
 - Sistemas de tempo real: múltiplos acessos a base de dados

Concorrência

- Principais problemas:

- sincronização (e.g., perda de sinais)



- exclusão mútua (e.g., bases de dados)



- bloqueios (espera infinita): deadlocks

Escalonamento de processos

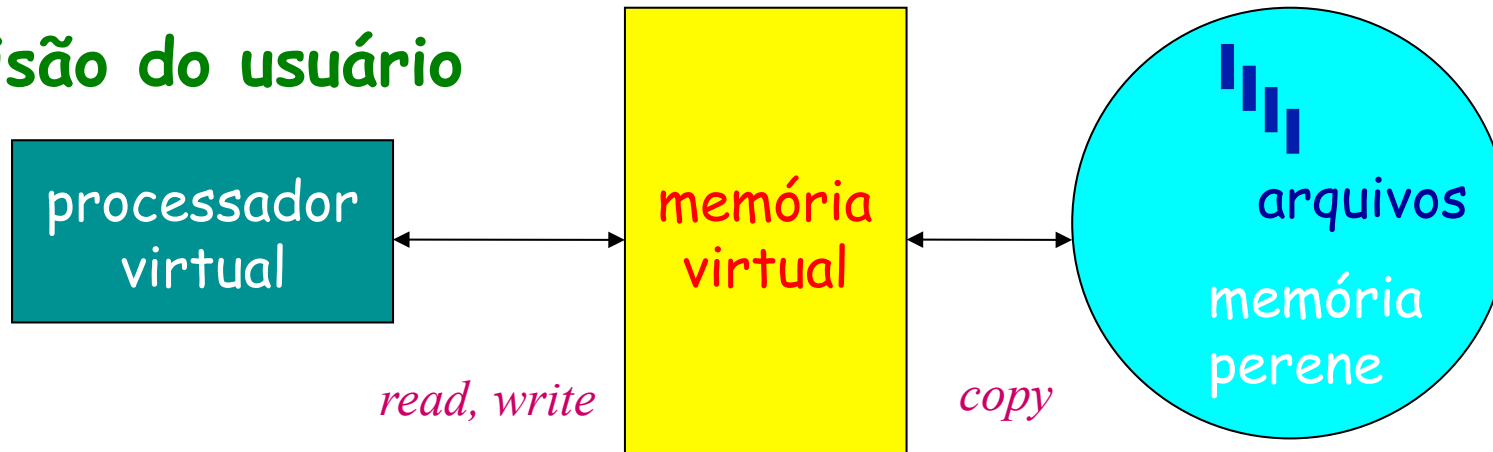
- Como escolher qual processo ocupará o processador?
- Alguns critérios:
 - justeza (*fairness*)
 - mas prioridades diferentes
 - diferenciação entre classes
 - tempos de resposta diferentes
 - eficiência
 - vazão máxima
 - minimizar tempo de resposta
- Níveis de escalonamento

Gerenciamento de memória

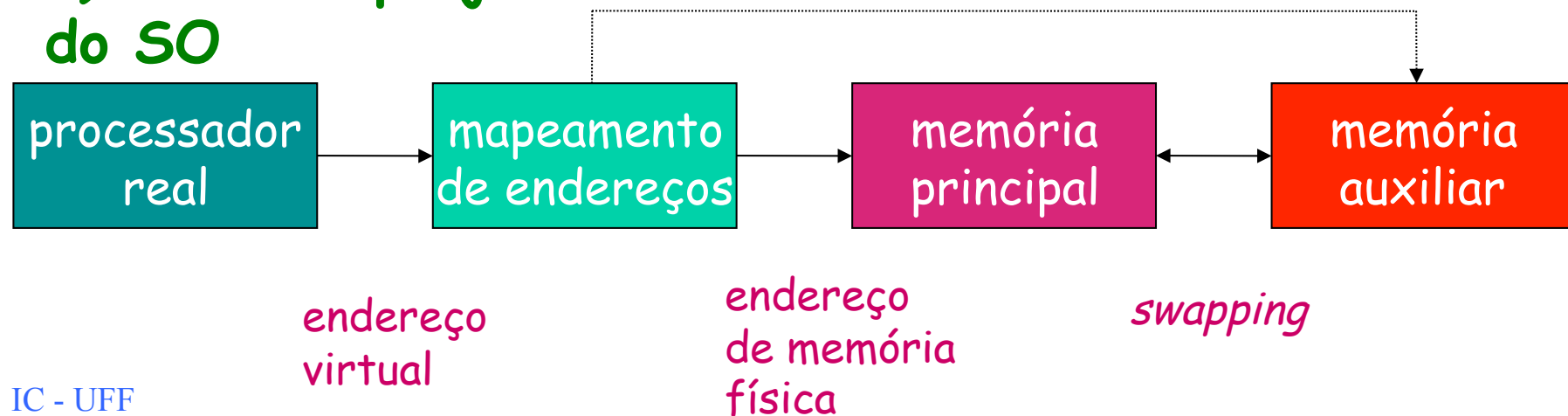
- Devido ao compartilhamento da MP
- Requisitos:
 - Gerenciar de acordo com hierarquia de memória
 - Isolação/proteção da área de MP entre processos
 - Demandas dinâmicas: módulos, procedimentos e área de dados
 - Ex.: pilha de memória
 - Proteção e controle de acesso
 - Ex.: áreas compartilhadas entre processos
 - Armazenamento permanente
- Solução: memória virtual + sistema de arquivos

Memória virtual?

i) visão do usuário



ii) visão do projetista do SO



Segurança e proteção

- Uma grande preocupação hoje em dia. O que está envolvido?
 - controle de acesso: quem pode acessar sistema e dados?
 - controle de fluxo de informação: quem pode receber o que
 - certificação: como saber quem é quem?

Formas de estruturação

- SOs cada vez mais complexos. Para estruturar, só a programação modular não é suficiente
 - estruturação em níveis
 - arquitetura em micro-núcleo
 - *threads e multithreads*

Formas de estruturação

- Em sistemas grandes
 - Camadas hierárquicas
 - abstração de informações
 - Cada camada contém funções de mesma complexidade, dimensão e abstração
 - Um nível maior utiliza funções de um nível abaixo
 - Os níveis de funcionalidade coincidem com os níveis de um sistema computacional

Formas de estruturação

- **Nível 1** - componentes
 - SO limpa registradores, acessa célula de memória
- **Nível 2** - instruções da máquina
 - SO executa instruções L2 para executar serviços
- **Nível 3** - lida com procedimentos e subrotinas
- **Nível 4** - tratamento de interrupções
 - Parte por software para salvar contexto

Formas de estruturação

- **Nível 5** - manipulação de processos
 - Rotinas de suspensão, escalonamento, sincronização, semáforos, ...
- **Nível 6** - manipulação de dispositivos de memória secundária
 - Leitura e gravação, manipulação de cabeçote
- **Nível 7** - manipulação de memória virtual
 - Transferência entre MP e MS
 - Por ex., utiliza nível 6
- **Nível 8** - gerenciamento de compartilhamento de informação e troca de msg's entre processos

Formas de estruturação

- **Nível 9** - manipulação de armazenamento secundário
 - Mais alto nível que 6
- **Nível 10** - interfaces para acesso a dispositivos externos
- **Nível 11** - rotinas de associação de identificadores de processos externos (usuários) e internos (endereços)

Outras formas

- Multiprocessamento simétrico
 - cada processador executa cópia do SO
- SOs distribuídos
 - fornece a ilusão de uma única memória principal
- Sistemas móveis

Outros requisitos

- Sistemas de tempo real (TR)
 - normalmente usados em aplicações dedicadas
 - requisitos temporais bem definidos
 - sistemas TR críticos
 - vale a pena usar memória virtual?
 - sistemas TR não-críticos
- Consumo de energia

Características Atuais

- Melhor tecnologia → melhor h/w → melhor s/w
 - **SO mais elaborado**: lida com redes, maior MP, multimídia, web, computação cliente-servidor
- Microkernel - mínimo de funções essenciais
 - Gerenciador de espaço de gerenciamento
 - Comunicação entre processos
 - Escalonamento
- Multithread - aplicação composta de vários threads

| Threads | Processos |
|--|---|
| <ul style="list-style-type: none">● unidade de um processo● Compartilha informações | <ul style="list-style-type: none">● Visão somente de suas variáveis● Comunicação troca msg |

Exercícios - Cap II

- 2.1 até 2.5

2.1) Suponha um computador multiprogramado, em que os processos têm características semelhantes. Em um dado período de computação, T , considerando cada processo, metade do tempo é gasto em E/S e outra metade em processamento. Cada processo ainda precisa de N períodos para ser executado. Assuma uma prioridade round-robin e que as operações de E/S possam ser sobrepostas com operações de processamento. Defina:

- a) turnaround de cada processo = tempo total para completar o processo
- b) $\frac{\text{Vazão/Throughput}}{T}$ = média de processos finalizados por período
- c) Utilização de processador = % de tempo que cada processador está ativo (não está esperando)

Calcule (a), (b) e (c) para 1, 2 e 4 processos simultâneos, assumindo que o período de tempo T é distribuído da seguinte maneira:

- a) Primeira metade = E/S e segunda metade processamento
- b) E/S durante o 1º e 4º quartos de tempo, e processamento, 2º e 3º quartos de tempo