

### Cap 19 - Estruturas de memória

→ A memória principal é um componente fundamental de qualquer sistema. Nela são mantidos os processos, threads e bibliotecas compartilhadas, além do núcleo do sistema operacional, com seu código e suas estruturas de dados.

\* A velocidade, custo e consumo de energia de cada tipo de memória é inversamente proporcional à sua capacidade.

#### → Memória Física

\* A quantidade de memória RAM disponível no computador constitui seu espaço de memória física.

\* Hoje em dia programas não operam sobre a memória física, em vez disso são usados sistemas de espelhos de endereçamento e memória virtual, a fim de tornar mais simples e flexível o uso de memória por processos e pelo SO.

do de espaço de endereçamento.

#### → Memória virtual

\* Para ocultar a organização complexa da memória física e simplificar os procedimentos de alocação de memória aos programas, é implementada a memória virtual, diferenciando endereços físicos de endereços lógicos.

↳ Físicos: são endereços da memória física do computador, definidos pela quantidade de memória disponível na máquina.

↳ Lógicos: são endereços usados pelos programas e pelo SO, definidos de acordo com o espaço de endereçamento do processador.

\* Nesta forma ao executor, processos somente possuem memória virtual, que é gerada pelo processador, estes endereços ao serem acessados são traduzidos em endereços físicos pelo MMU, que por motivos de desempenho é integrado ao chip da CPU, funcionando independente.

\* A MMU possui e implementa diferentes estratégias para a tradução de endereços: por segmentos, por partições e por páginas.

→ Partições

\* Consiste em dividir a memória física em  $N$  partições, que podem ter tamanhos variados, em cada partição é salvado um ~~proteção~~.

→ Por exemplo: o espaço ocupado por uma partição é de  $T$  bytes, então os intervalos lógicos vão de  $[0, T-1]$  bytes, a MMU guarda dois registradores, a base, que é o endereço físico inicial da partição, e o limite, que define o número de bytes máximo da partição.

\* A tabela de partições reside na memória R/M, onde é usada para atualizar os registradores de base e limite da MMU.

\* A melhor vantagem é a simplicidade, por depender apenas de dois registradores é simples de implementar até em hardware de baixo custo.

## → Segmentos

É uma extensão das partições, porém neste caso cada parte de um processo possui espaço de endereçamento próprio, cada uma destas áreas é denominada segmento, e possui seus próprios reguladores de base e limite. Cada processo possui sua Tabela de Segmentos, que possui as informações de cada segmento. Quanto o número de segmentos seja pequeno, a tabela de segmentos pode residir no processador, caso o número seja maior, deve-se manter armazenada na RAM.

## Cap 15 - Uso da memória

### Cap 15 - Uso da memória

- a) O núcleo não faz parte do espaço de endereçamento dos processos, possui um pequeno conjunto de páginas do núcleo ativas, para receber respostas e interrupções, o restante fica numa Tabela de páginas separada.
- b) Cada processo é implementado pelo SO como uma síndula, ou seja, uma área de memória que temente ele e o núcleo possuem acesso, que contém informações como código binário, variáveis, bibliotecas, etc.
- c) Em linguagens como C e C++, variáveis podem ser alocadas de duas formas:
  - 1) Alocação estática: o espaço necessário para a variável é definido durante a compilação do programa, o espaço da memória RAM é definido no início da execução e mantido até seu encerramento. Em C, são usadas as variáveis globais e variáveis locais estáticas.

↳ alocação automática: são as variáveis definidas dentro de uma função (variáveis locais e parâmetros), são alocadas na pilha de execução do programa, são alocadas quando a função é invocada, e liberadas quando a função termina.

↳ alocação dinâmica: são variáveis em que o programador aloca explicitamente blocos de memória para armazenar dados, os utiliza e depois os libera.

---

Cap 16 - Alocação de memória.

