MC514–Sistemas Operacionais: Teoria e Prática

1s2009

Gerenciamento de Memória

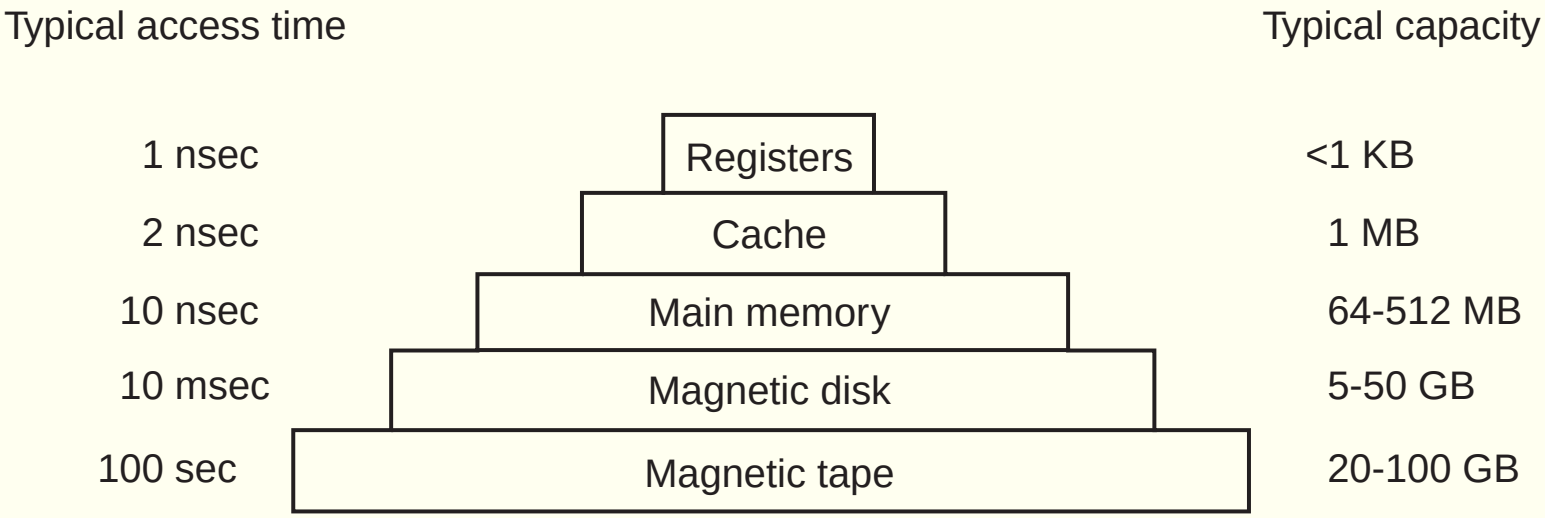
Sonho dos usuários

Idealmente, a memória deveria ser

- rápida,
- de custo baixo,
- imensa e
- não volátil.

Realidade para os usuários

Hierarquia de Memória



Tanenbaum: Figura 1.7

Como estão estes valores hoje?

Hierarquia de Memória

Registradores

- Internos à CPU
- Extremamente rápidos
- Otimizações de código podem mover temporariamente variáveis para registradores.
- Programas podem dar palpites sobre o que deve ficar armazenado nos registradores

```
register int r;
```

Hierarquia de Memória

Cache

- Internos ou muito próximos à CPU
- Divididos em linhas de cache
- Controlados por hardware
- Cache hit
- Cache miss

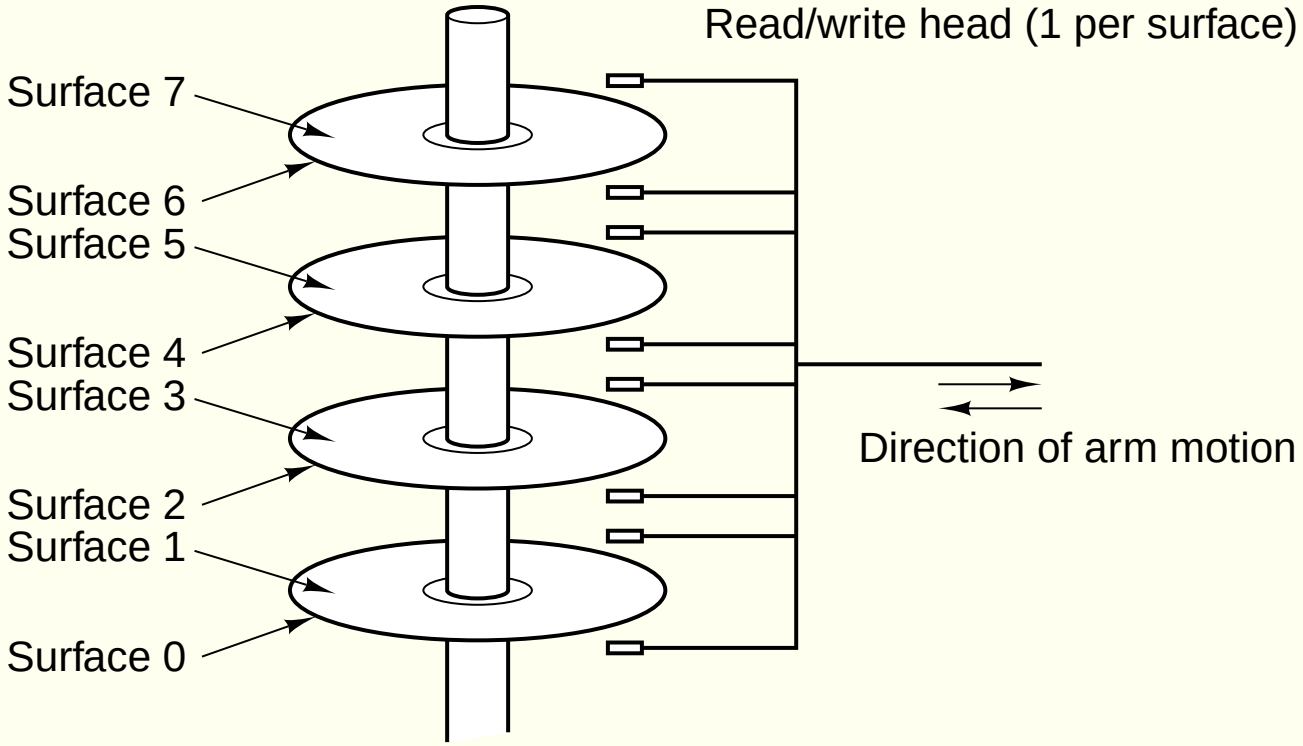
Hierarquia de Memória

Memória Principal

- Random Access Memory (RAM)
- Compromisso entre preço e desempenho
- Armazenamento volátil

Hierarquia de Memória

Disco



Hierarquia de Memória

Fitas magnéticas

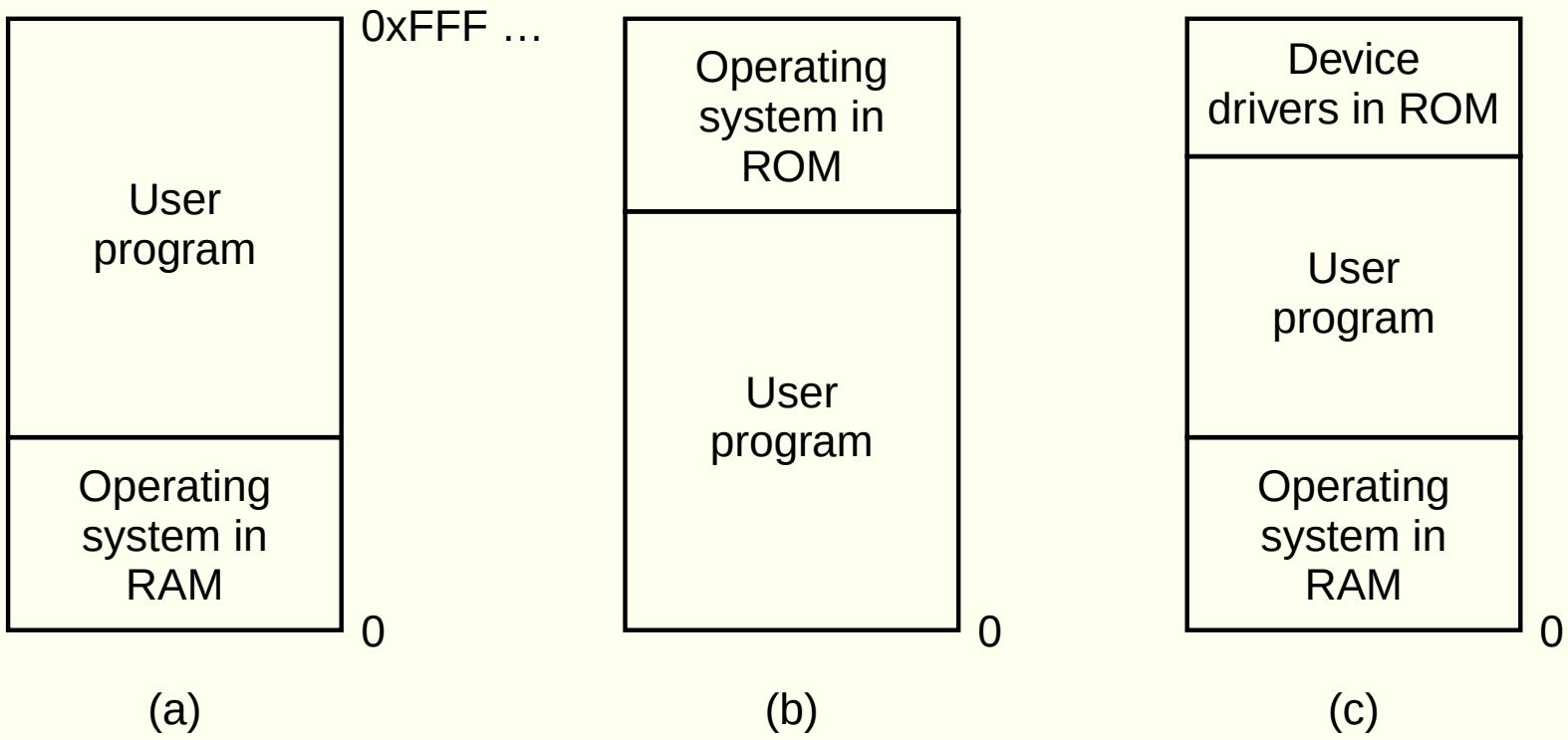
- Utilizadas para cópias de segurança (backups)
- Armazenamento de grandes quantidades de dados
- Acesso seqüencial

Hierarquia de Memória

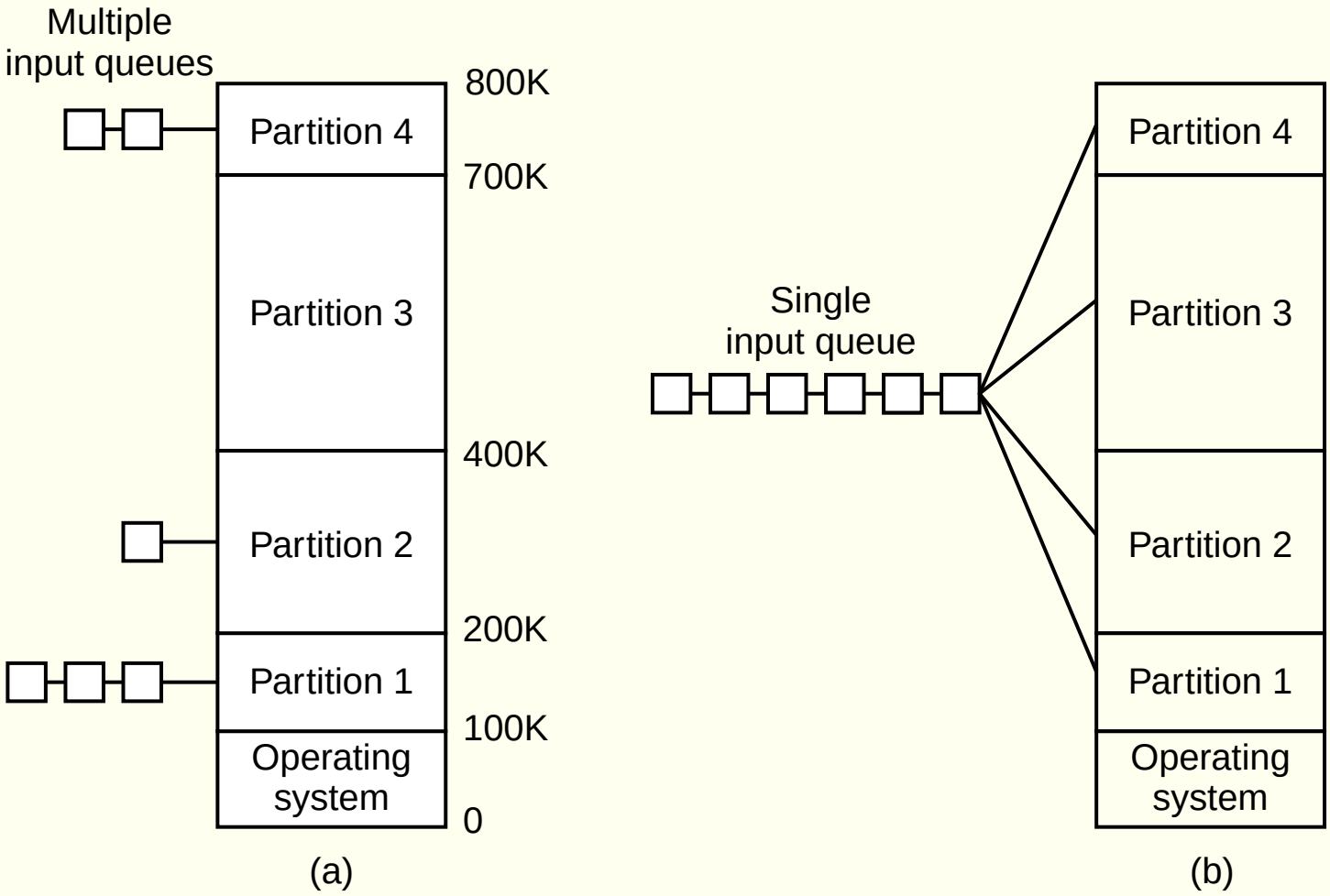
Outros tipos de memória

- ROM (Read Only Memory)
  - rápida e barata
  - bootstrap loader está gravado em ROM
- EEPROM (Electrically Erasable ROM)
  - podem ser apagadas (erros podem ser corrigidos)
- CMOS
  - dependem de uma bateria
  - armazenam relógio e configurações

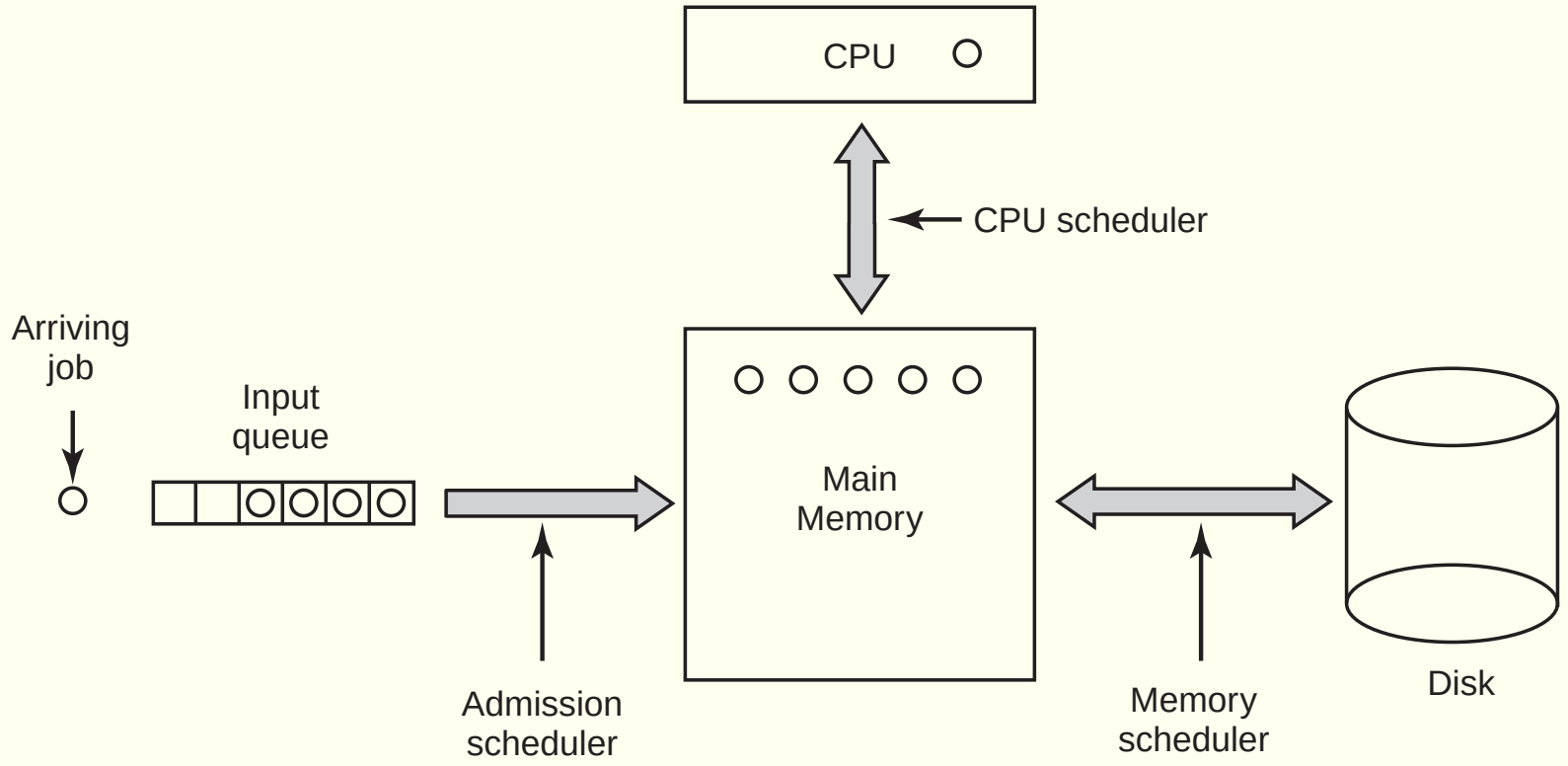
Monoprogramação



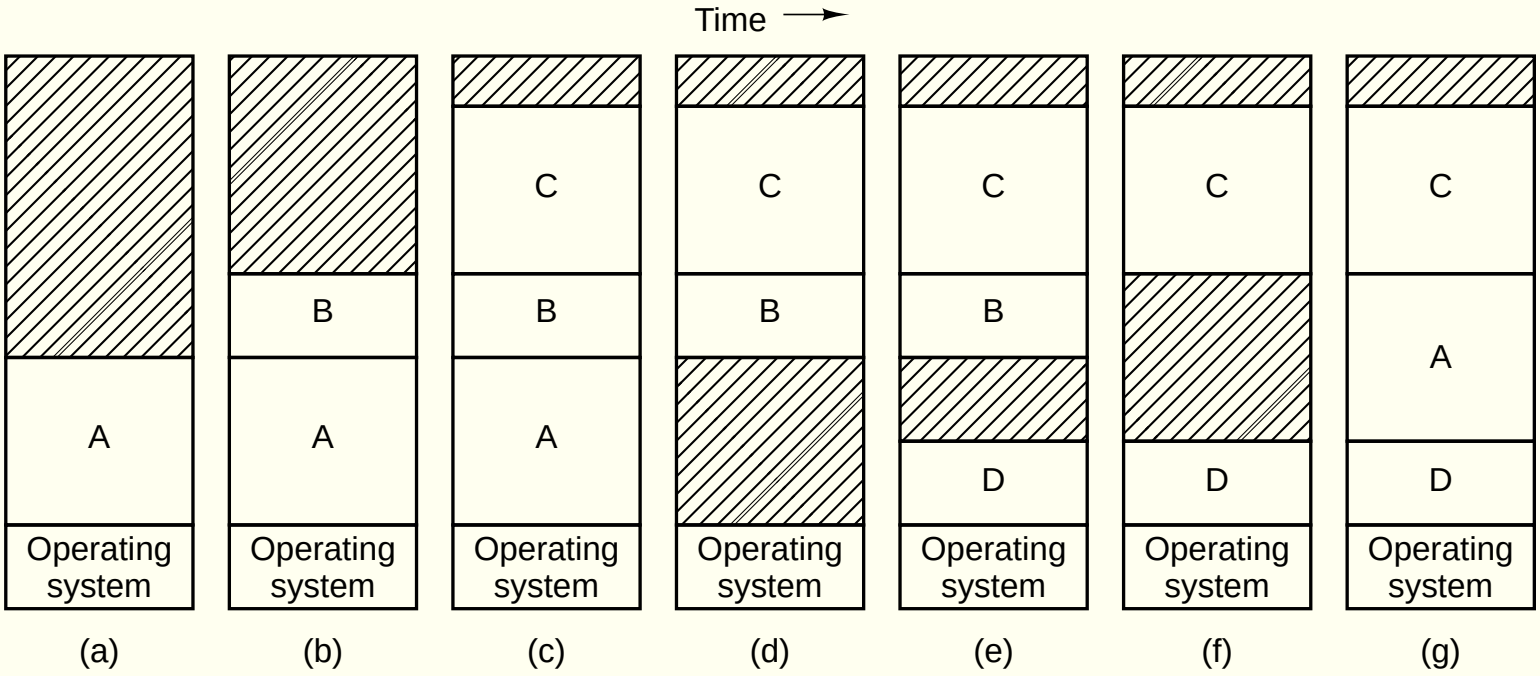
Multiprogramação e partições fixas



Swapping


CPU
CPU scheduler
Arriving job
Input queue
Main Memory
Admission scheduler
Memory scheduler
Disk

Swapping



Relocação e Proteção

- Relocação: um programa deve poder rodar em endereços físicos distintos.
- Proteção: um programa não pode fazer acesso à área de memória reservada a outro programa.

Relocação durante a carga

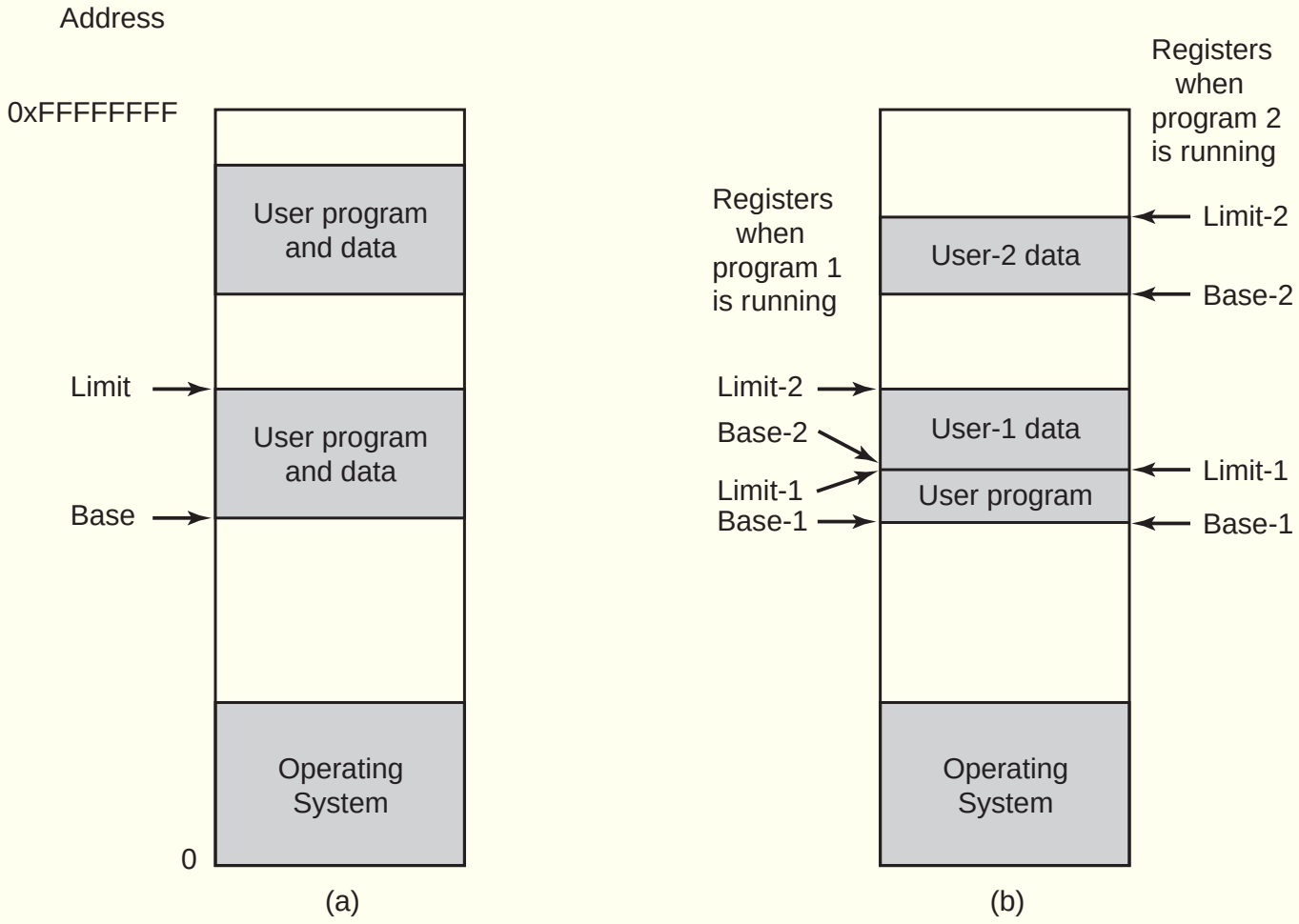
- Todos os endereços precisam ser identificados e alterados
- Não resolve o problema da proteção

Bits de proteção

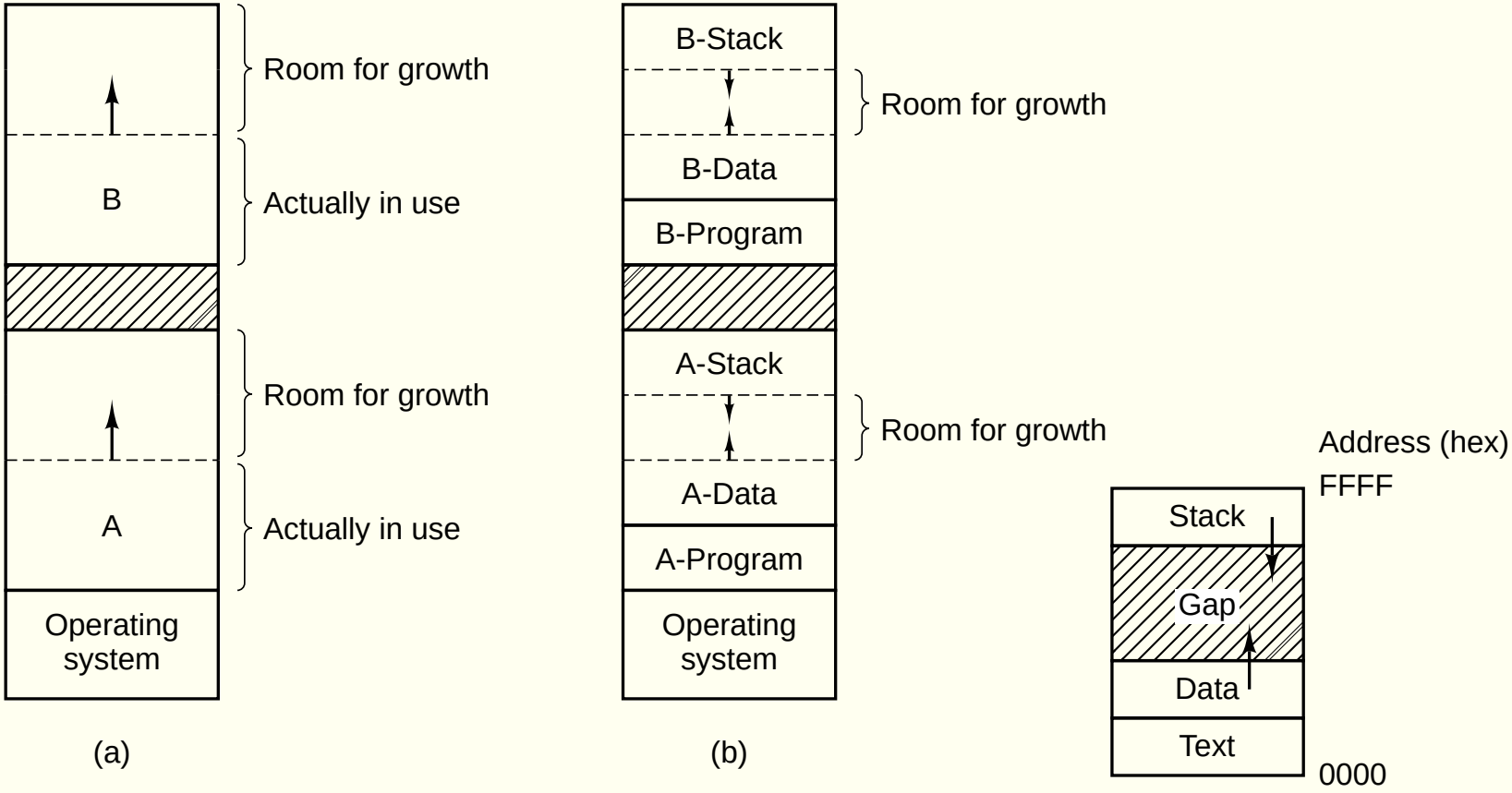
1010	
1010	
1010	
0011	
0011	

O PSW de cada processo deve conter os bits de proteção

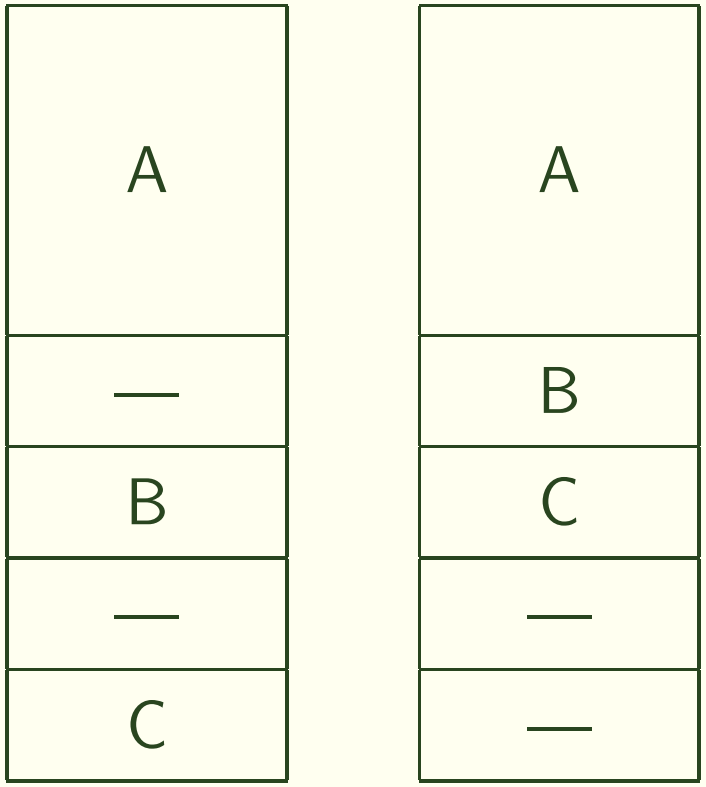
Registradores base e limite



Espaço para crescimento



Compactação de memória



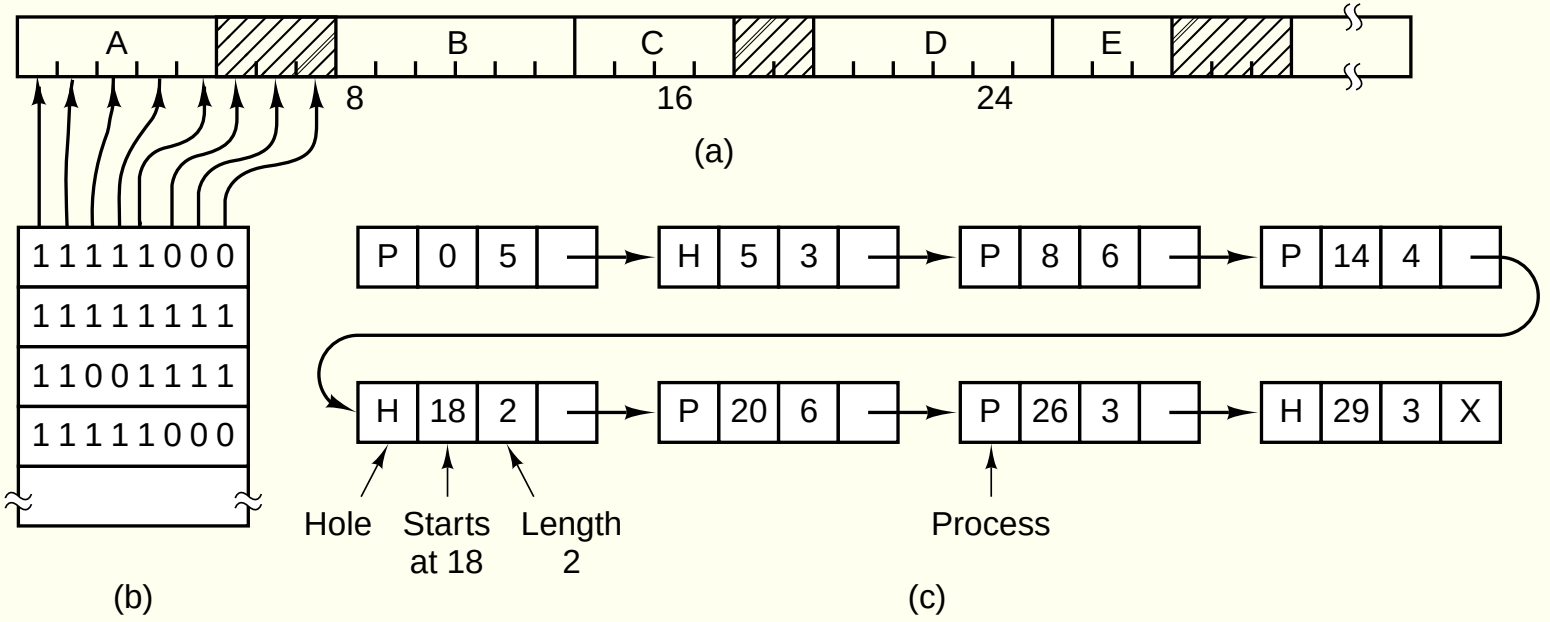
Gerência de grandes blocos de memória

Problema relacionado

Malloc, free e realloc

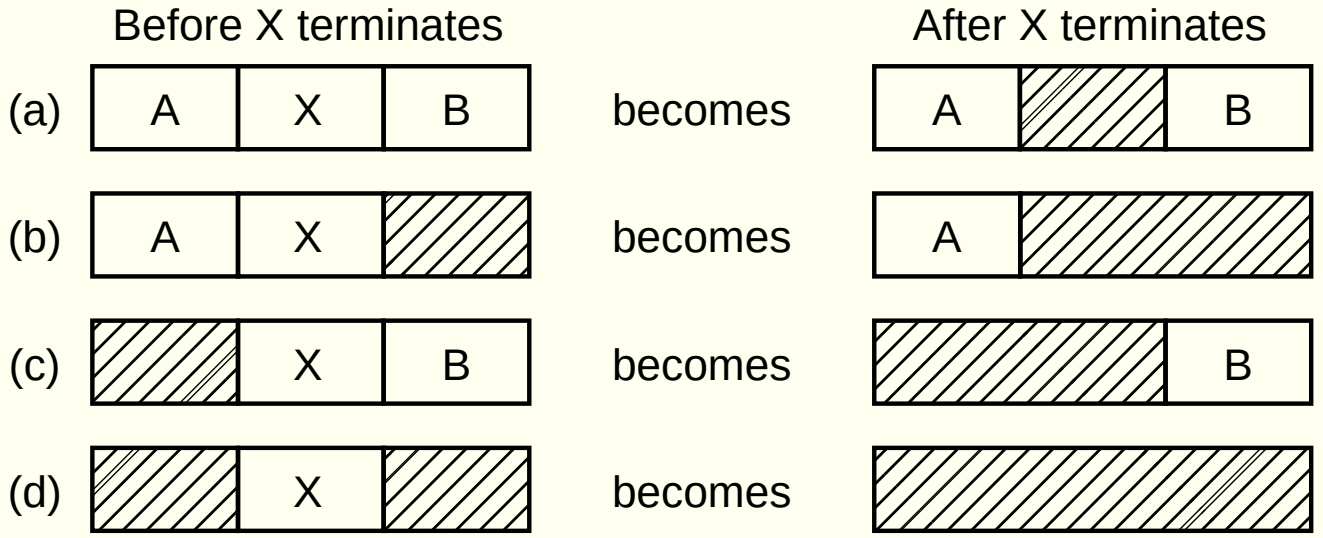
```
void *malloc(size_t size);
void free(void *ptr);
void *realloc(void *ptr, size_t size);
```

Bitmaps e lista de livres



Tanenbaum: Figura 4.7

Atualização da lista



Tanenbaum: Figura 4.8

Veja o código: erro_free.c

Algoritmos para alocação de memória

- *First fit*
- *Next fit*
- *Best fit*
- *Worst fit*
- Veja o código fit.c

Programas muito grandes

O que fazer se um programa for muito grande para caber na memória?

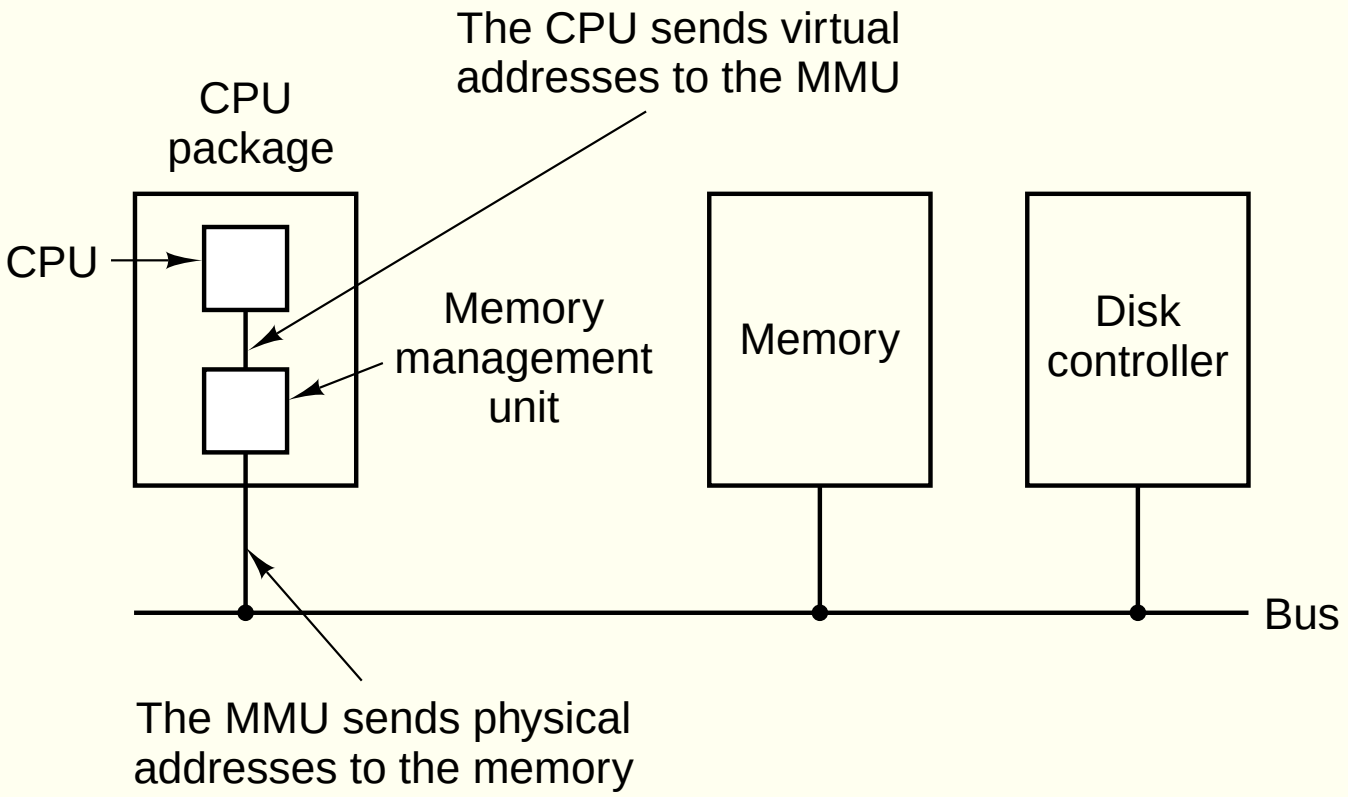
Overlays

```
dados d1, d2, d3, d4, d5;

f1();    g1();   h1();
f2();    g2();   h2();
f3();    g3();   h3();

main() {
  fase_1();  /* funçoes f e dados d1, d2, d3 */
  fase_2();  /* funçoes g e dados d1, d2, d4 */
  fase_3();  /* funçoes h e dados d1, d2, d5 */
  }
```

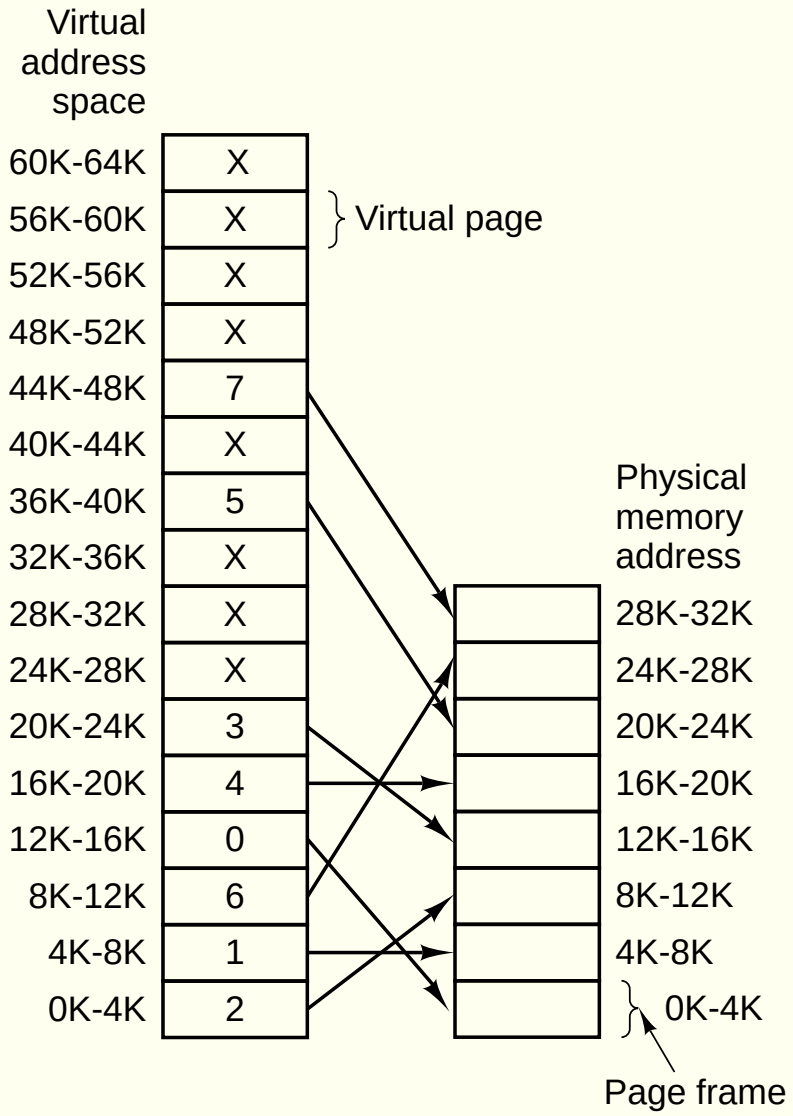
Memória Virtual



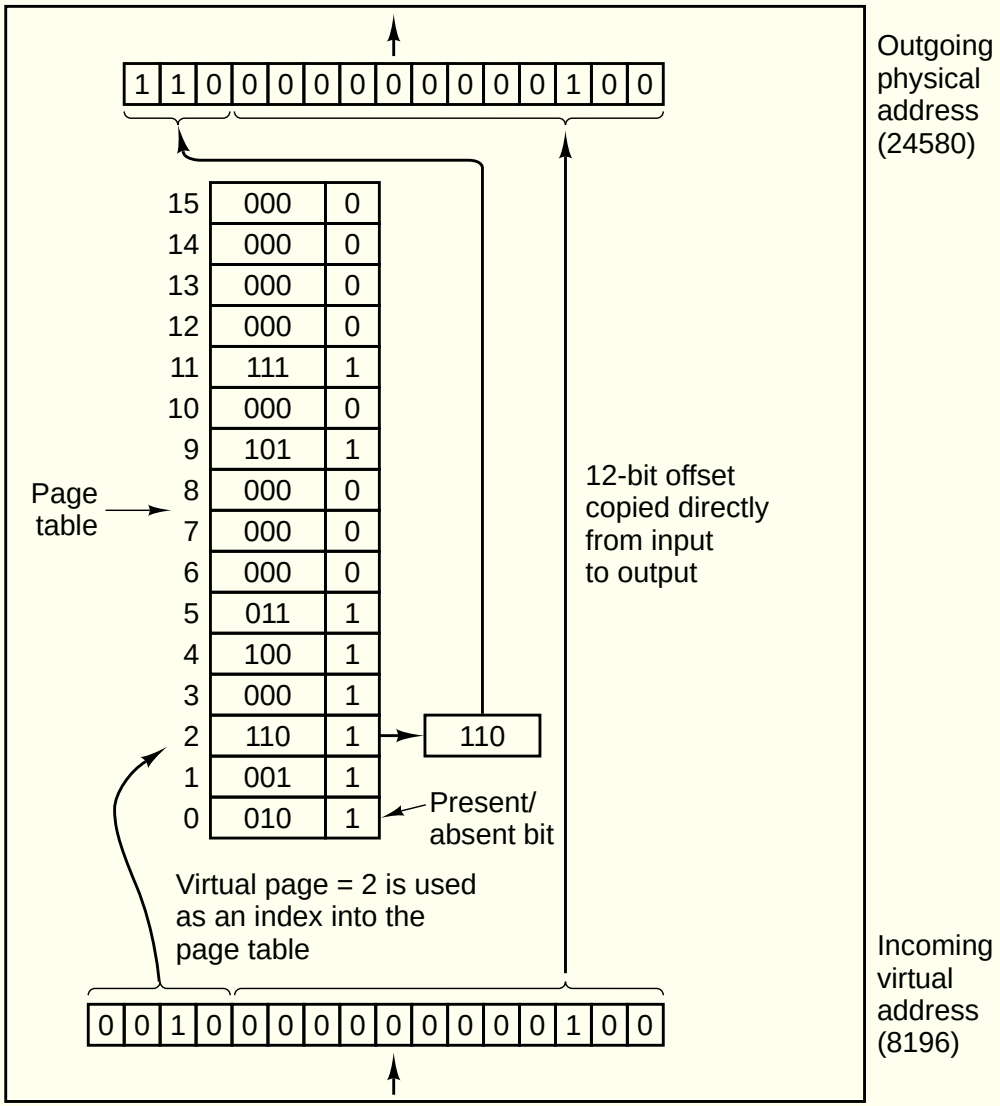
Tanenbaum: Figura 4.9

Memory Management Unit (MMU)

Paginação


Virtual address space
60K-64K
56K-60K
52K-56K
48K-52K
44K-48K
40K-44K
36K-40K
32K-36K
28K-32K
24K-28K
20K-24K
16K-20K
12K-16K
8K-12K
4K-8K
0K-4K
X
X
X
X
7
X
5
X
X
X
3
4
0
6
1
2
Virtual page
Physical memory address
28K-32K
24K-28K
20K-24K
16K-20K
12K-16K
8K-12K
4K-8K
0K-4K
Page frame

Mapeamento dos endereços



Paginação - Exemplo

- 32 bits de endereço
- páginas de 4k
- 20 primeiros bits indicam a página
- 12 últimos bits indicam o deslocamento dentro da página

- Veja o código pagesize.c

Espaço de endereçamento

- Apenas as páginas ocupadas precisam ser mapeadas
- Veja o código sbrk.c