

Organização de Computadores

Memória Virtual

Background

- Técnica que permite a execução de processos que não podem estar completamente na memória física.
- Instruções devem estar na memória
 - Soluções:
 - carregar todo programa
 - utilizar overlays e dynamic loading
 - Problema
 - esforço maior do programador

Background

- Constatação:
 - na maioria do tempo, os programas não são utilizados por completo
 - código para condições não usuais
 - matrizes, lista e tabelas grandes
 - opções utilizadas raramente
- Mesmo o programa todo sendo necessário, não é usado em um mesmo momento.
 - Benefícios :
 - Programas escritos para espaço de endereço grande.
 - Mais programas podem ser executados

Endereços Físicos e Virtuais

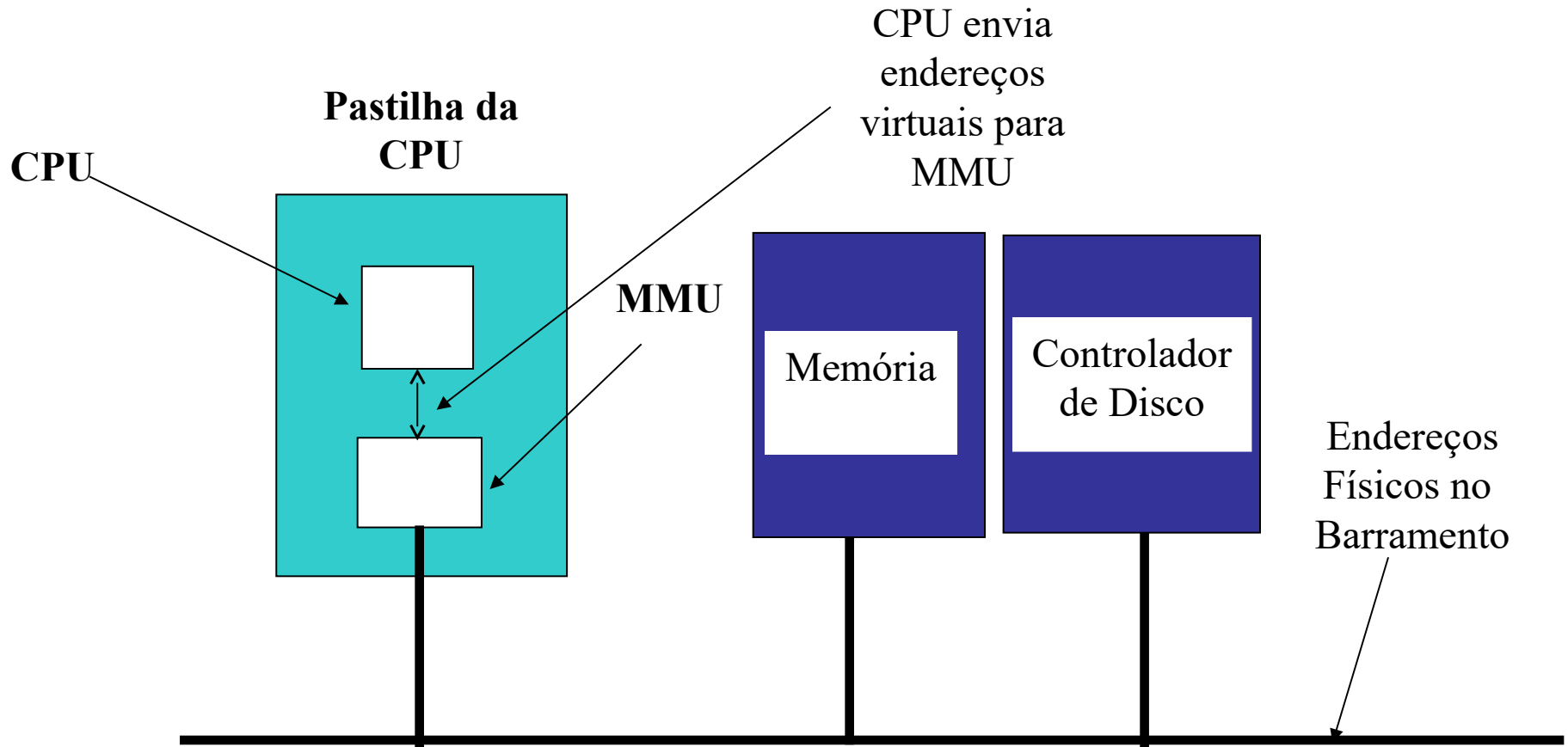
- O programa é escrito como se a máquina possuísse uma memória “gigante”.
- Nas Instruções de Máquina do programa sempre estão presentes *Endereços Virtuais*.
- É a CPU quem converte o *Endereço Virtual* (V) da Instrução no *Endereço Físico* (I)

mov AX, [0FF0:3258]



Endereço Virtual

ESQUEMA FÍSICO



Implementação da MV

- O espaço de Memória Virtual é dividido em porções de tamanho fixo denominados *páginas*.
- Uma página, pode estar na memória ou no *disco*.
- A maioria das páginas está no disco, apenas as páginas sendo utilizadas estão na memória. *Elas tem sempre o mesmo tamanho.*
- O *quadro de página* é uma página presente na memória.

Memória Virtual
(64K)

- 0-4K
- 4K-8K
- 8K-12K
- 12K-16K
- 16K-20K
- 20K-24K
- 24K-28K
- 28K-32K
- 32K-36K
- 36K-40K
- 40K-44K
- 44K-48K
- 48K-52K
- 52K-56K
- 56K-60K
- 60K-64K

2
1
6
0
4
3
X
X
X
5
X
7
X
X
X
X

Memória Física (32K)

- 0-4K
- 4K-8K
- 8K-12K
- 12K-16K
- 16K-20K
- 20K-24K
- 24K-28K
- 28K-32K

Páginas de 4K

Páginas de 4K
X - Página está no disco
(página virtual)
N - Página na memória
(page frame)

MMU

- A MMU mapeia o endereço virtual para o page frame se a página estiver na memória ou causa um *trap* para o SO se a página estiver no disco.
- É necessário portanto uma estrutura (*page table*) que diga para um dado endereço virtual qual é o número de seu page frame ou se a página está no disco (não há page frame).

mov reg, 8192 (página virtual 2) (começa do 0)



mov reg, 24576 (página física 6) (começa do 0)

Neste caso está na memória

MMU

mov reg, 32780 (página virtual 8) (começa do 0)



mov reg, ??? Não está na memória

1) O sistema operacional pega uma página menos usada e escreve seu conteúdo de volta para o disco.

Exemplo pega o *page frame 1* e escreve de volta para o disco.

2) O sistema operacional pega uma página do disco e traz para a memória.

Exemplo: Coloca a página do disco posição de memória física 4K (correspondente ao *page frame 1*).

Falta de Página

...

or AL, AL

jz 012F

push SI

push DI

call 2255

jnb 0139

mov CX, [0104]

cld

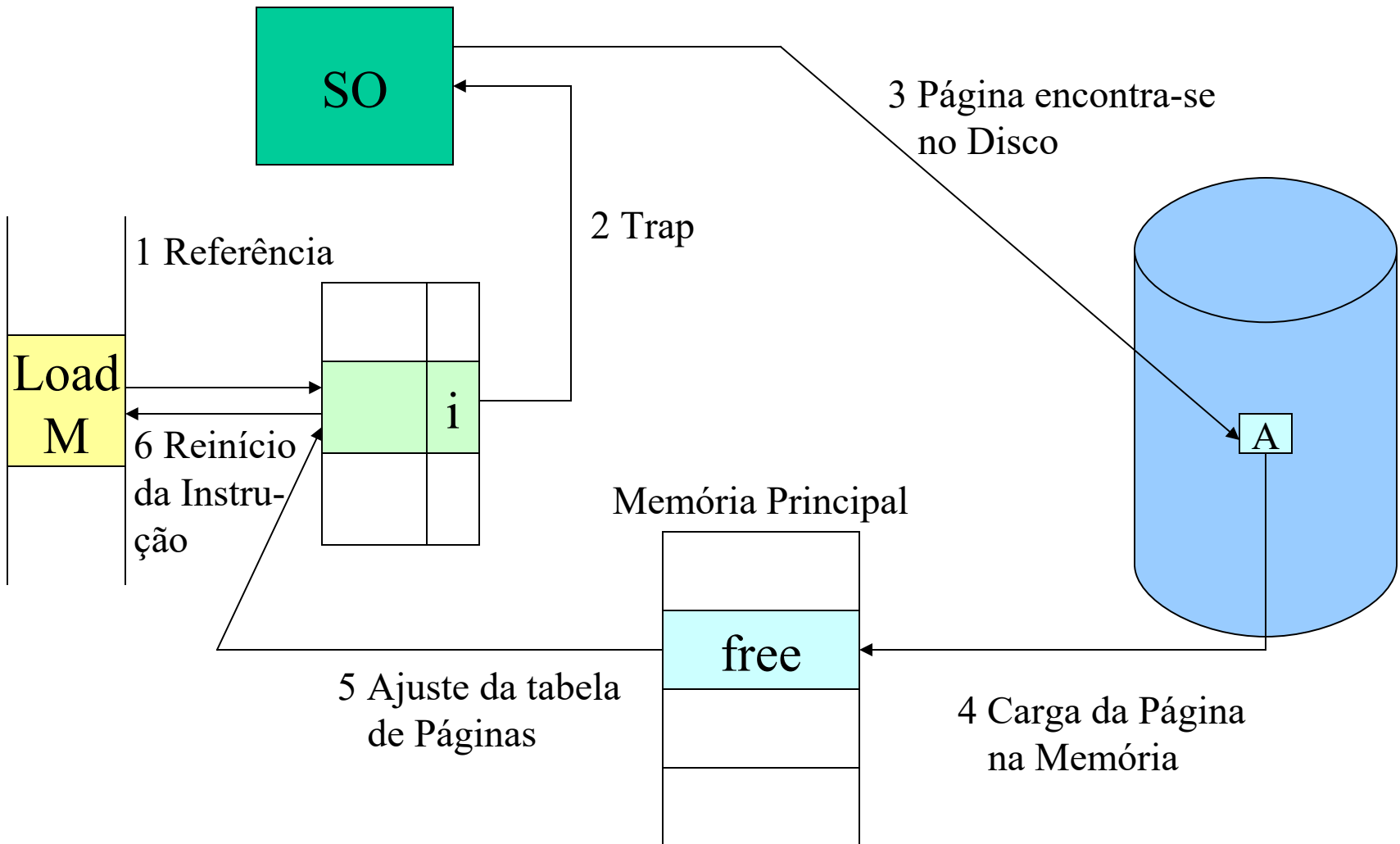
push SI

push DI

...

- Supor que o endereço 0104h referencia uma página que não está na memória.
- Se a página estiver na memória, o programa continua normalmente.
- Se a página não estiver na memória acontece a falta de página (*page fault*).

Falta de Página



Memória Virtual
(64K)

- 0-4K
- 4K-8K
- 8K-12K
- 12K-16K
- 16K-20K
- 20K-24K
- 24K-28K
- 28K-32K
- 32K-36K
- 36K-40K
- 40K-44K
- 44K-48K
- 48K-52K
- 52K-56K
- 56K-60K
- 60K-64K

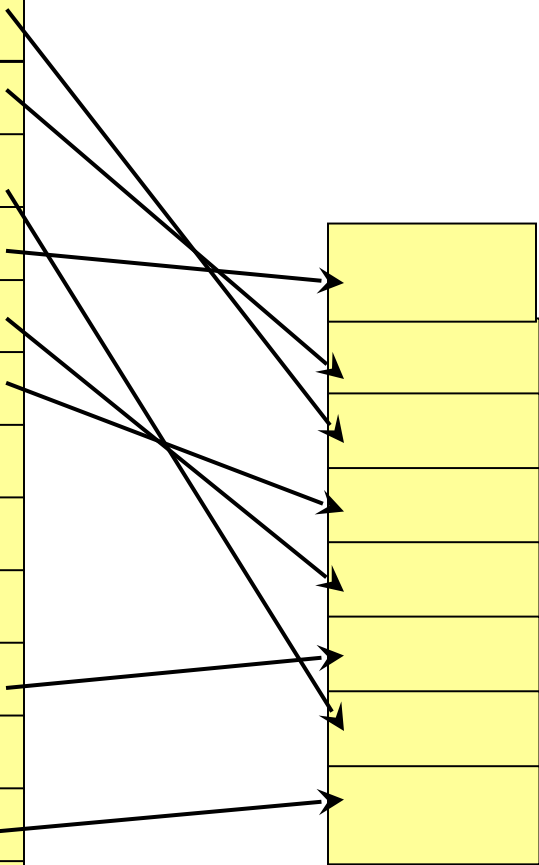
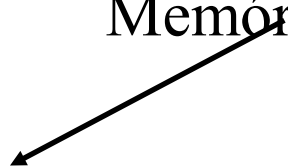
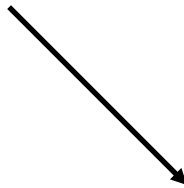
2
1
6
0
4
3
X
X
X
5
X
X
X
X
X
X

Memória Física (32K)

Páginas de 4K

- 0-4K
- 4K-8K
- 8K-12K
- 12K-16K
- 16K-20K
- 20K-24K
- 24K-28K
- 28K-32K

CPU quer ler
endereço 32780



Memória Virtual
(64K)

- 0-4K
- 4K-8K
- 8K-12K
- 12K-16K
- 16K-20K
- 20K-24K
- 24K-28K
- 28K-32K
- 32K-36K
- 36K-40K
- 40K-44K
- 44K-48K
- 48K-52K
- 52K-56K
- 56K-60K
- 60K-64K

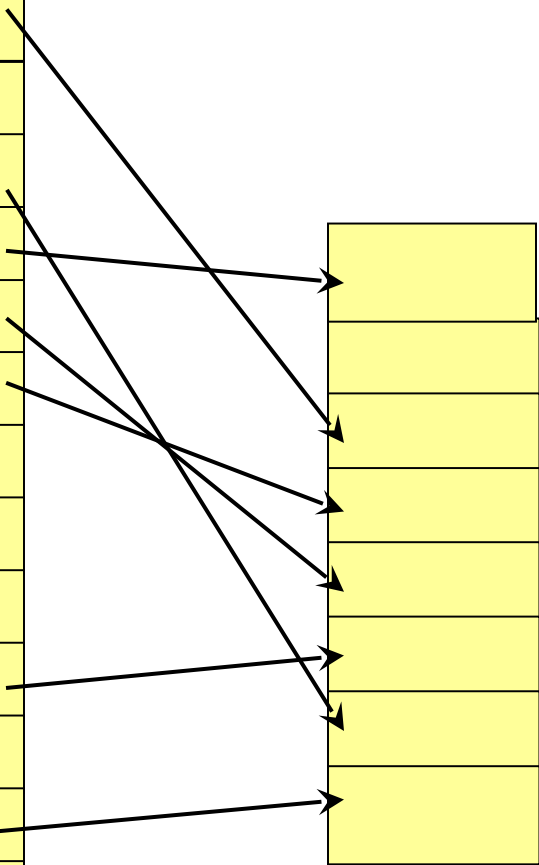
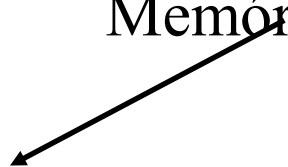
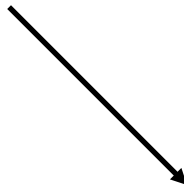
2
1
6
0
4
3
X
X
X
5
X
X
X
X
X

Memória Física (32K)

Páginas de 4K

- 0-4K
- 4K-8K
- 8K-12K
- 12K-16K
- 16K-20K
- 20K-24K
- 24K-28K
- 28K-32K

CPU quer ler
endereço 32780



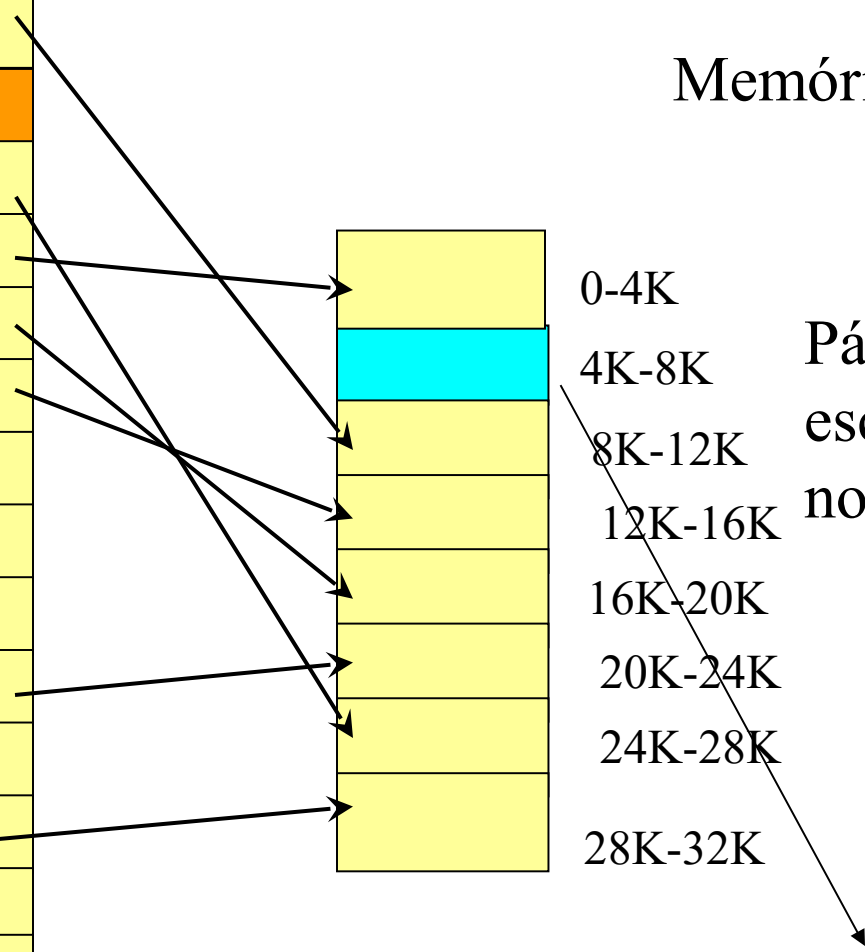
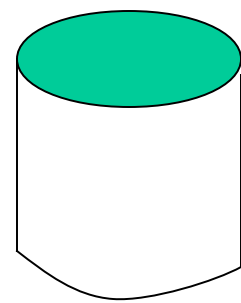
Memória Virtual
(64K)

0-4K	2
4K-8K	X
8K-12K	6
12K-16K	0
16K-20K	4
20K-24K	3
24K-28K	X
28K-32K	X
32K-36K	X
36K-40K	5
40K-44K	X
44K-48K	7
48K-52K	X
52K-56K	X
56K-60K	X
60K-64K	X

Memória Física (32K)

0-4K
4K-8K
8K-12K
12K-16K
16K-20K
20K-24K
24K-28K
28K-32K

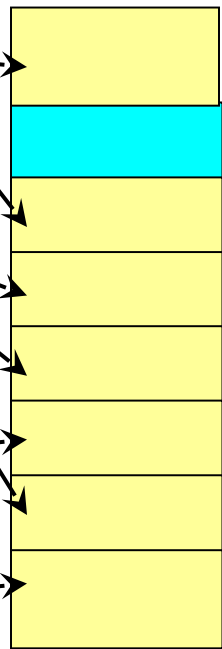
Página 1 é
escrita de volta
no disco.



Memória Virtual
(64K)

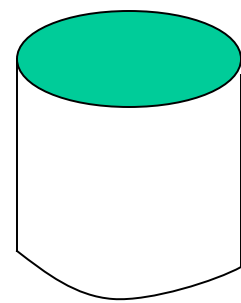
0-4K	2
4K-8K	X
8K-12K	6
12K-16K	0
16K-20K	4
20K-24K	3
24K-28K	X
28K-32K	X
32K-36K	X
36K-40K	5
40K-44K	X
44K-48K	7
48K-52K	X
52K-56K	X
56K-60K	X
60K-64K	X

Memória Física (32K)



- 0-4K
- 4K-8K
- 8K-12K
- 12K-16K
- 16K-20K
- 20K-24K
- 24K-28K
- 28K-32K

Página Virtual 8
(endereço **32780**) é
carregada no
page frame 2.



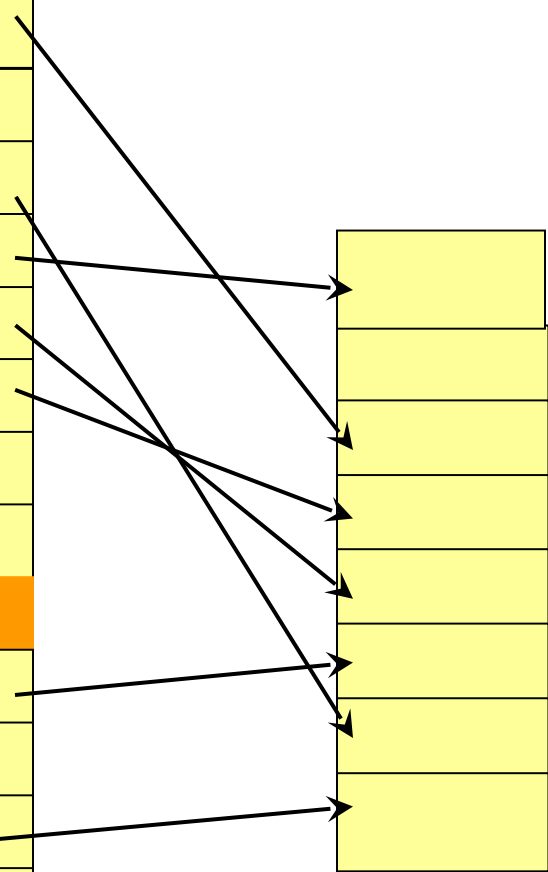
Memória Virtual
(64K)

0-4K	2
4K-8K	X
8K-12K	6
12K-16K	0
16K-20K	4
20K-24K	3
24K-28K	X
28K-32K	X
32K-36K	X
36K-40K	5
40K-44K	X
44K-48K	7
48K-52K	X
52K-56K	X
56K-60K	X
60K-64K	X

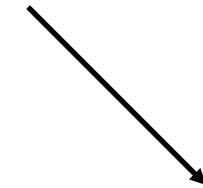
Memória Física (32K)

0-4K
4K-8K
8K-12K
12K-16K
16K-20K
20K-24K
24K-28K
28K-32K

Página Virtual 8
(endereço
32780) é
carregada no
page frame 2.

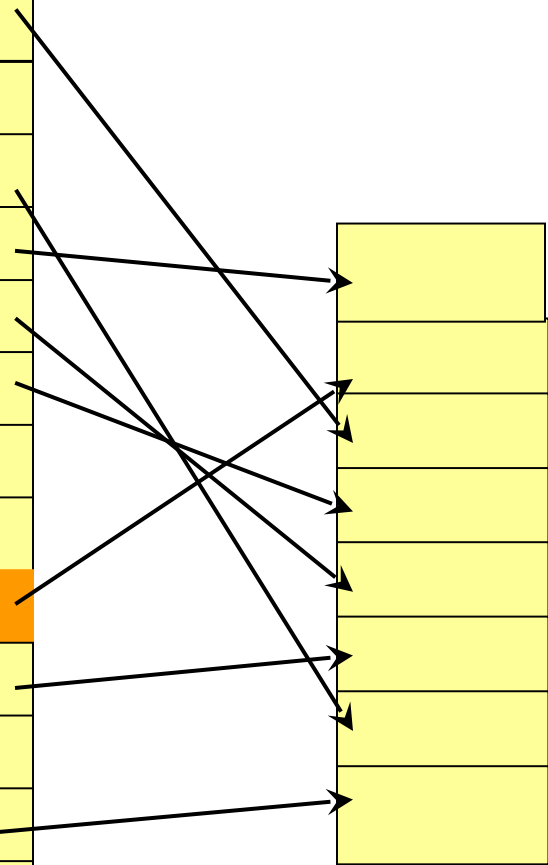


Memória Virtual
(64K)



0-4K
4K-8K
8K-12K
12K-16K
16K-20K
20K-24K
24K-28K
28K-32K
32K-36K
36K-40K
40K-44K
44K-48K
48K-52K
52K-56K
56K-60K
60K-64K

2
X
6
0
4
3
X
X
1
5
X
7
X
X
X



Memória Física (32K)

0-4K
4K-8K
8K-12K
12K-16K
16K-20K
20K-24K
24K-28K
28K-32K



Page Table é ajustado

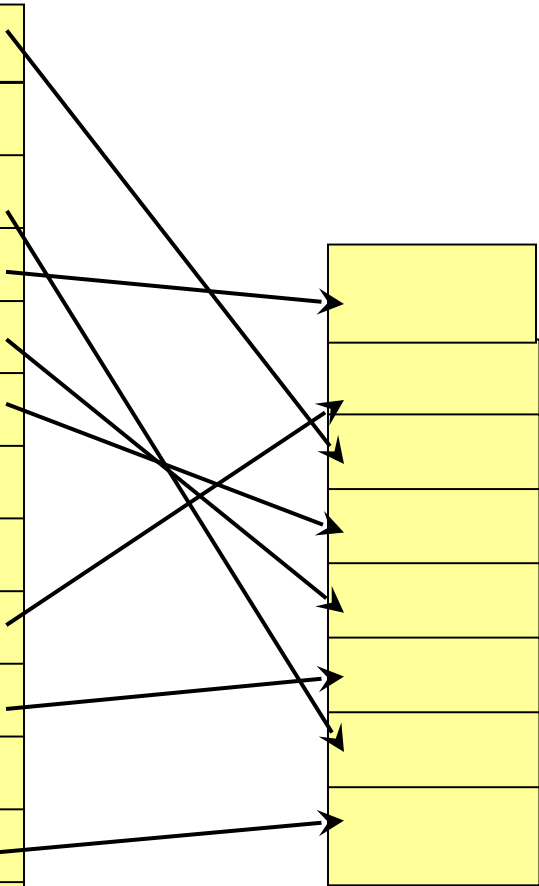
Memória Virtual
(64K)

- 0-4K
- 4K-8K
- 8K-12K
- 12K-16K
- 16K-20K
- 20K-24K
- 24K-28K
- 28K-32K
- 32K-36K
- 36K-40K
- 40K-44K
- 44K-48K
- 48K-52K
- 52K-56K
- 56K-60K
- 60K-64K

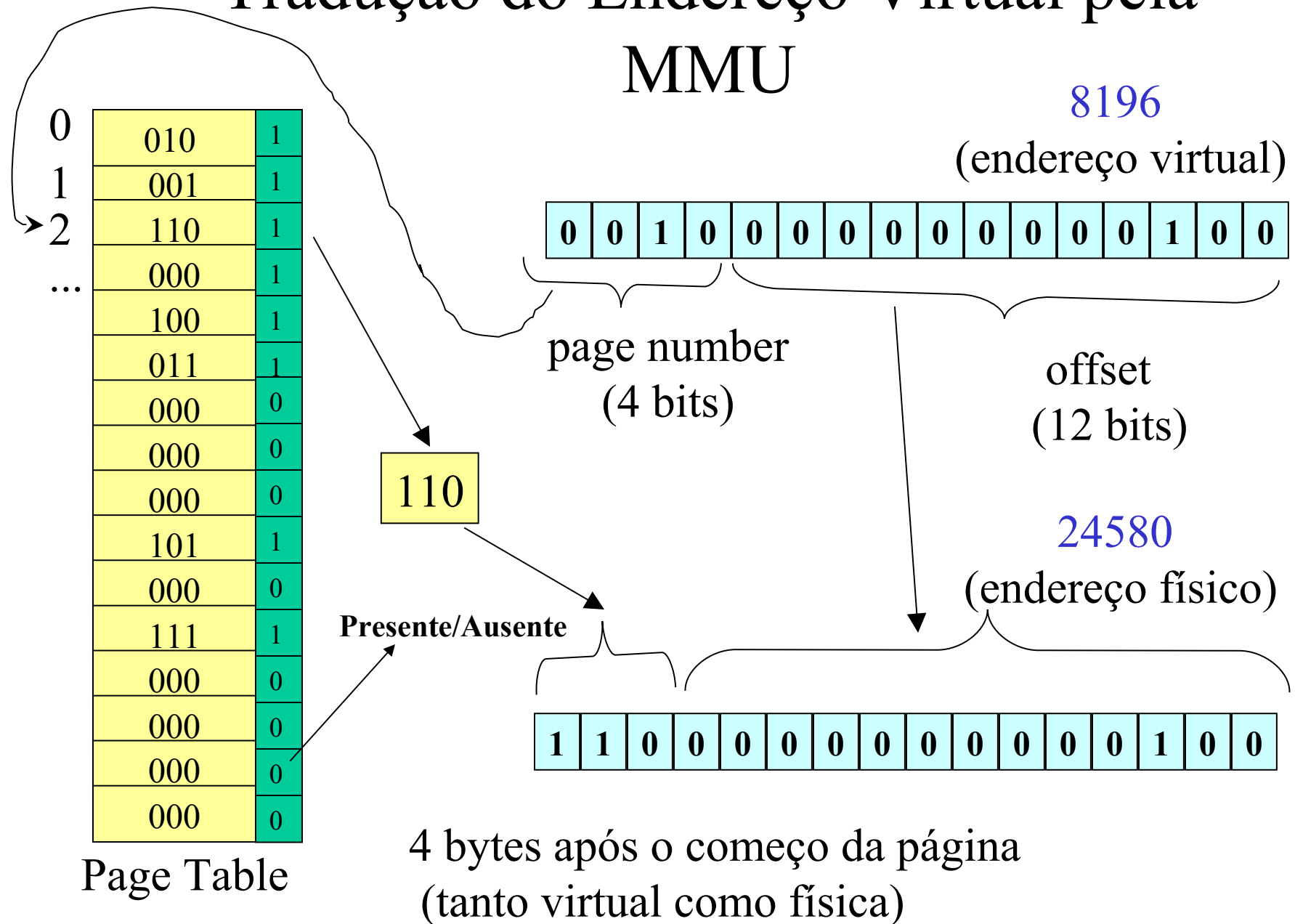
2
X
6
0
4
3
X
X
1
5
X
X
X
X
X

Memória Física (32K)

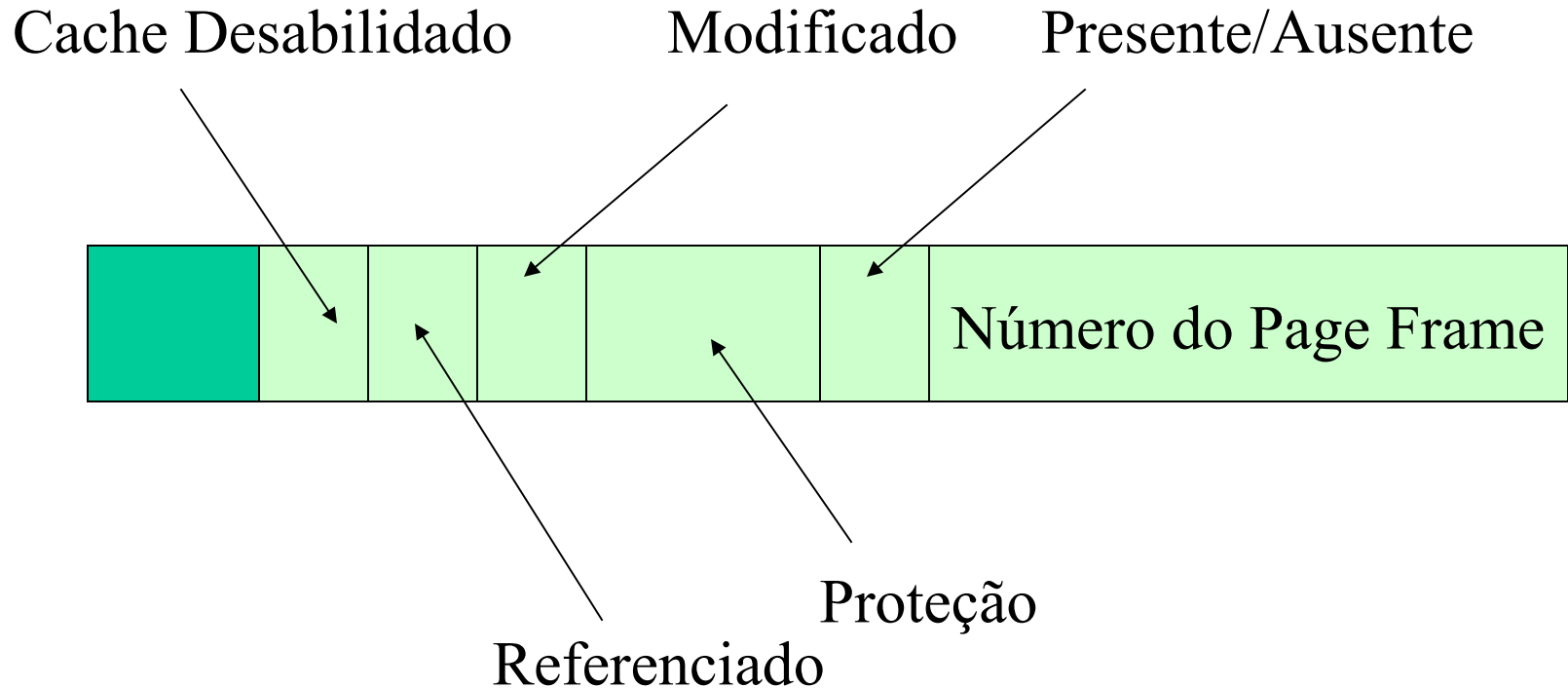
- 0-4K
- 4K-8K
- 8K-12K
- 12K-16K
- 16K-20K
- 20K-24K
- 24K-28K
- 28K-32K



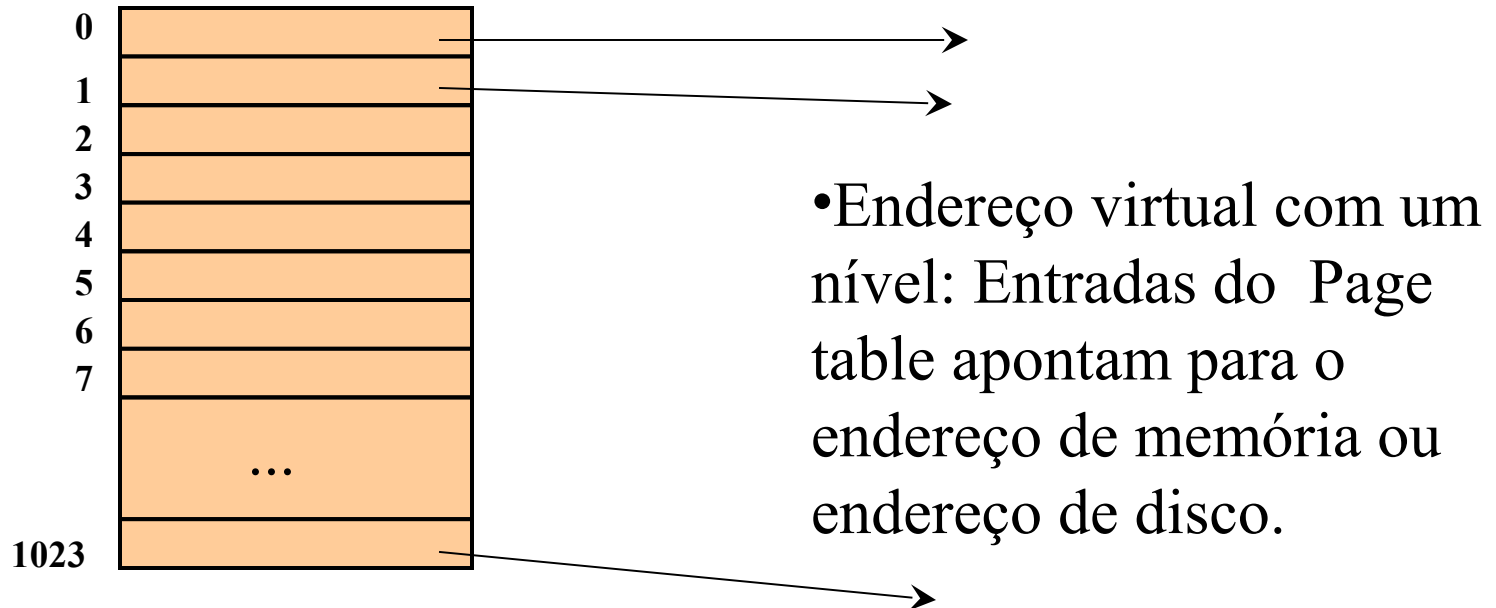
Tradução do Endereço Virtual pela MMU



Page Table Típico



Esquema do Page Table



PT1	Offset
-----	--------

Endereço Virtual.

Exemplo de Tabela de Páginas

m	a	w	c	o	haddr	hdaddr
---	---	---	---	---	-------	--------

m - 1 Página está na memória

0 Página está no disco

a - 1 Página foi acessada

0 Página não foi acessada

w - 1 Página foi escrita

0 Página não foi escrita

c - 1 Página de código

0 Página de dados

o - 1 Página execute only

0 Página execute/read.

haddr -endereço da página na memória
(se a página estiver na memória).

hdhaddr - endereço da página no disco
(se a página estiver no disco, isto é
trilha, cabeça e setor).

Falta de Página

- Trap para SO
- Armazenamento dos registradores e estados
- Determinação que é uma falta de página
- Teste de referência e determinação do local no disco
- Leitura em um frame livre
- Enquanto espera, escalona-se processos
- Interrupção do disco
- Armazenamento dos registradores e troca de contexto
- Determinação de interrupção de disco
- Correção da tabela de páginas
- Espera-se pela relocação do processo
- Continua a instrução interrompida

Interrupções de Software e de Hardware

Programa A

mov ax, 1000h

mov dx, 30h

mov cx, ax

dec cx

cmp cx, 0h

jz 0889h

mov ax, [2000]

out dx, al

jmp 0FF0h

Programa B

cmp ax, 2000h

jz 300h

mov cx, 10

dec cx

cmp cx, 8h

jz 0889h

mov ax, [2000]

dec ax

jz 0F80h

Programa C

mov ax, 100h

dec cx

mov ax, cx

dec ax

cmp ax, 0h

jz 0809h

int 13h

mov ah, 10

cmp dx, 20

Programa D

cmp cx, 1

jz 100h

cmp cx, 2

jz 150h

cmp cx, 3

jz 200h

mov ah, 10

cmp dx, 20

jnz 300h

Execução: A

Pronto: B C D

Bloqueado:

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: A

Pronto: B C D

Bloqueado:

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: A

Pronto: B C D

Bloqueado:

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: A

Pronto: B C D

Bloqueado:

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: A

Pronto: B C D

Bloqueado:

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: A

Pronto: B C D

Bloqueado:

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: A

Pronto: B C D

Bloqueado:

Evento: Page Fault

Ação: Bloquear o processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução:

Pronto: B C D

Bloqueado: A

Evento: Page Fault

Ação: Bloquear o processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: C D

Bloqueado: A

Evento: Page Fault

Ação: Escalar Processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: C D

Bloqueado: A

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: C D

Bloqueado: A

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: C D

Bloqueado: A

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: C D

Bloqueado: A

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: C D

Bloqueado: A

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: C D

Bloqueado: A

Evento: Time Slice

Ação: Suspend Processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução:

Pronto: C D B

Bloqueado: A

Evento: Time Slice

Ação: Suspend Processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: C

Pronto: D B

Bloqueado: A

Evento: Time Slice

Ação: Escalar Processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: C

Pronto: D B

Bloqueado: A

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: C

Pronto: D B

Bloqueado: A

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: C

Pronto: D B

Bloqueado: A

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: C

Pronto: D B

Bloqueado: A

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: C

Pronto: D B

Bloqueado: A

Evento: IntHW do Disco

Ação: Desbloquear Process

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: C

Pronto: D B A

Bloqueado:

Evento: Int HW do Disco

Ação: Desbloquear Process

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: C

Pronto: D B A

Bloqueado:

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: C

Pronto: D B A

Bloqueado:

Evento: IntSW do Disco

Ação: Bloquear Processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução:

Pronto: D B A

Bloqueado: C

Evento: IntSW do Disco

Ação: Bloquear Processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx,1
jz 100h
cmp cx,2
jz 150h
cmp cx,3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: D

Pronto: B A

Bloqueado: C

Evento: IntSW do Disco

Ação: Escalar Processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: D

Pronto: B A

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: D

Pronto: B A

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: D

Pronto: B A

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: D

Pronto: B A

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: D

Pronto: B A

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: D

Pronto: B A

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: D

Pronto: B A

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: D

Pronto: B A

Bloqueado: C

Evento: Time Slice

Ação: Suspend Processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução:

Pronto: B A D

Bloqueado: C

Evento: Time Slice

Ação: Suspend Processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: A D

Bloqueado: C

Evento: Time Slice

Ação: Escalar Processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: A D

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: A D

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: A D

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: A D

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: A D

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: A D

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: B

Pronto: A D

Bloqueado: C

Evento: Time Slice

Ação: Suspend Processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução:

Pronto: A D B

Bloqueado: C

Evento: Time Slice

Ação: Suspend Processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: A

Pronto: D B

Bloqueado: C

Evento: Time Slice

Ação: Escalar Processo

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: A

Pronto: D B

Bloqueado: C

Evento:

Ação:

Interrupções de Software e de Hardware

Programa A

```
mov ax, 1000h
mov dx, 30h
mov cx, ax
dec cx
cmp cx, 0h
jz 0889h
mov ax, [2000]
out dx, al
jmp 0FF0h
```

Programa B

```
cmp ax, 2000h
jz 300h
mov cx, 10
dec cx
cmp cx, 8h
jz 0889h
mov ax, [2000]
dec ax
jz 0F80h
```

Programa C

```
mov ax, 100h
dec cx
mov ax, cx
dec ax
cmp ax, 0h
jz 0809h
int 13h
mov ah, 10
cmp dx, 20
```

Programa D

```
cmp cx, 1
jz 100h
cmp cx, 2
jz 150h
cmp cx, 3
jz 200h
mov ah, 10
cmp dx, 20
jnz 300h
```

Execução: A

Pronto: D B

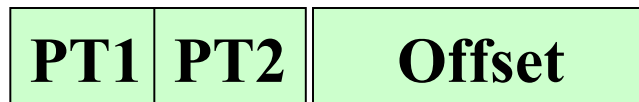
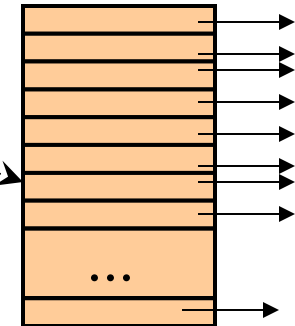
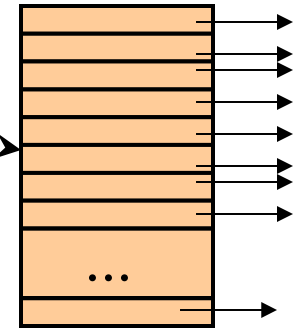
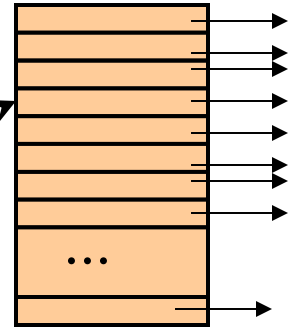
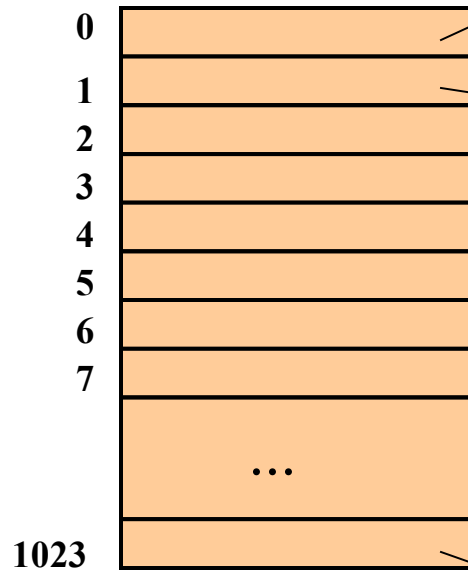
Bloqueado: C

Evento:

Ação:

Tabelas de Página Multinível

Entrada no
page table
aponta para
tabela da
página que
aponta para
memória



Endereço virtual.

Exemplos: 386,486, Pentium