

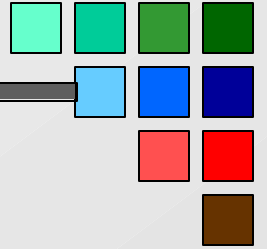
# Sistemas Distribuídos

**Ricardo Ribeiro dos Santos**

**[ricrs@ec.ucdb.br](mailto:ricrs@ec.ucdb.br)**

# Tópicos

---

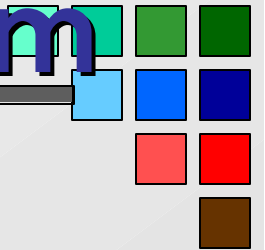


- Consistência e replicação em SD
- Motivação
- Consistência voltada para Dados

# Consistência e Replicação em SD

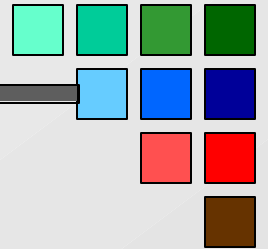
- Aumentar a confiabilidade e melhorar o desempenho
  - Confiabilidade:
    - Continuar o trabalho depois de um *crash*!
  - Desempenho:
    - Atender o crescente número de clientes e serviços
    - Replicação e *caching* são técnicas amplamente empregadas para melhorar o desempenho
- Problema:
  - Como manter as cópias consistentes?

# Consistência e Replicação em SD

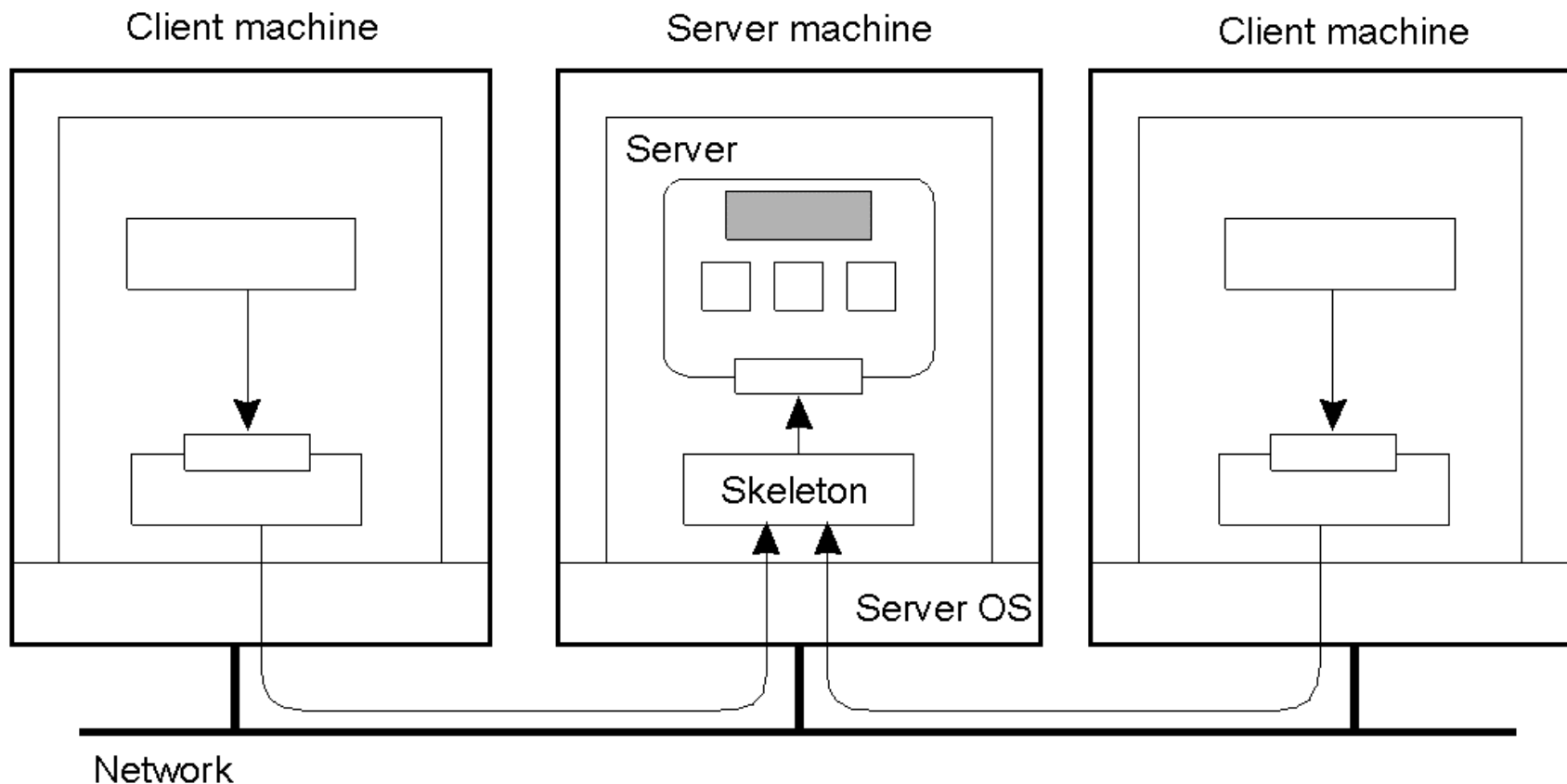


- Questões a serem tratadas:
  - Manter consistência requer maior utilização da banda de Rede
  - Necessidade de sincronização
- Um dilema:
  - Replicação e *caching* melhoram o desempenho. Por outro lado, manter todas as cópias consistentes requer sincronização que geralmente acaba por degradar o desempenho!

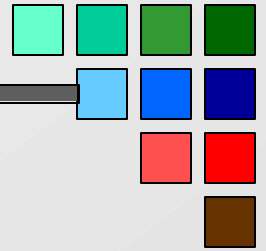
# Replicação de Objetos (1)



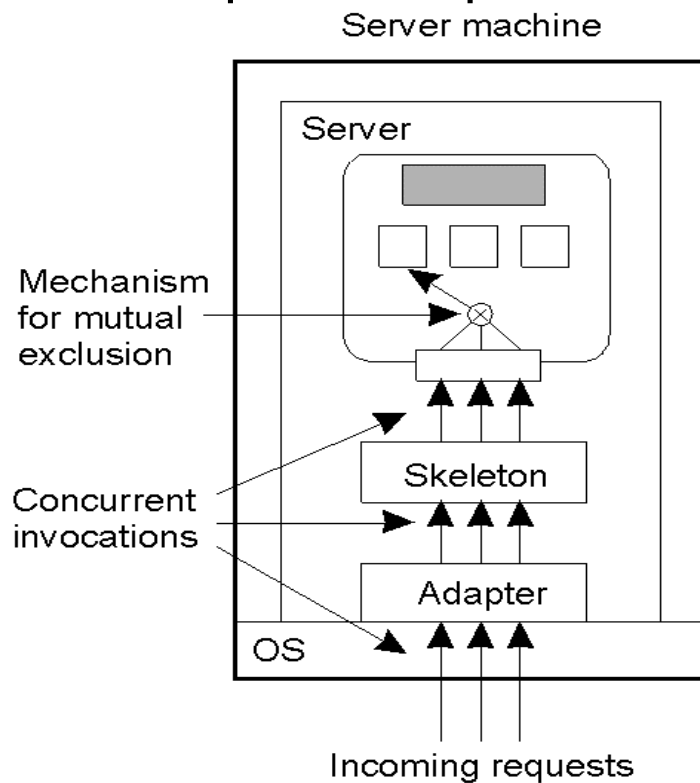
- Organização de um objeto distribuído compartilhado por dois diferentes clientes



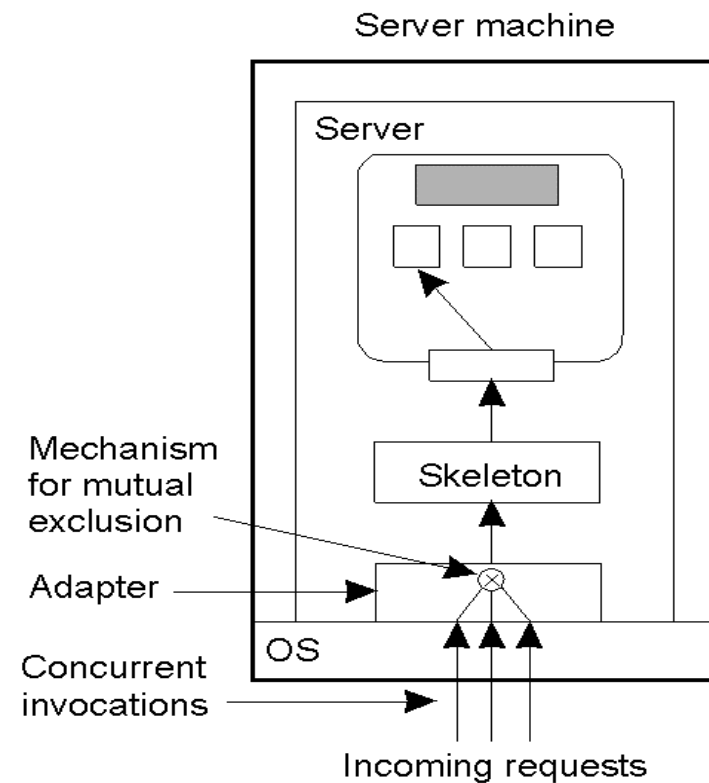
# Replicação de Objetos(2)



- Um objeto remoto capaz de manipular invocações concorrentes
- Um objeto remoto que requer um adaptador de objetos para manipular requisições concorrentes

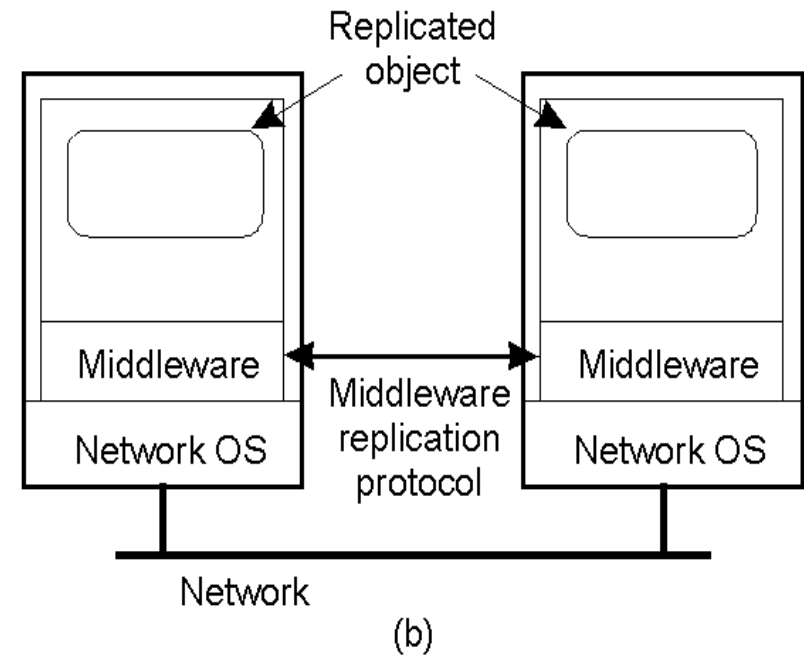
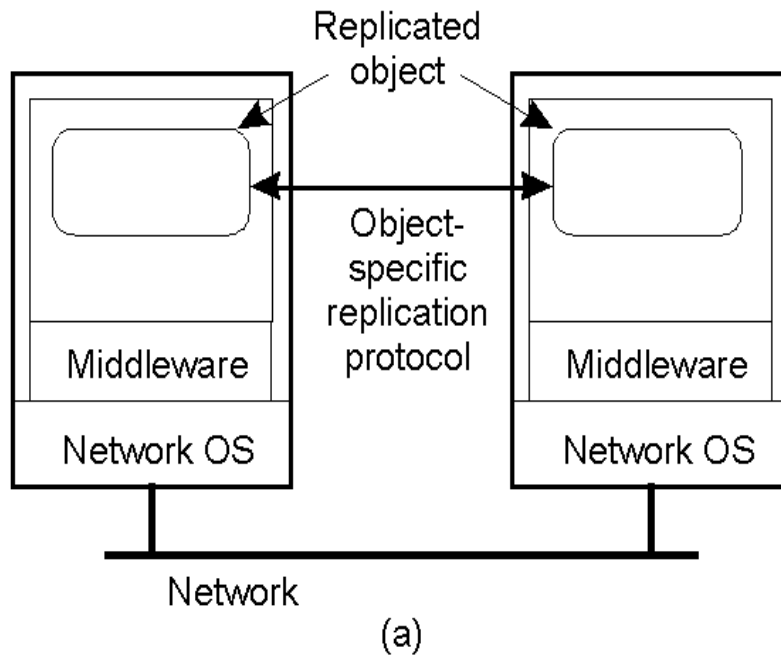
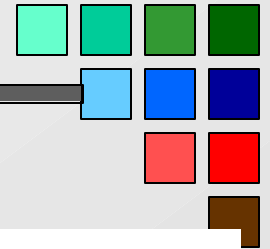


(a)



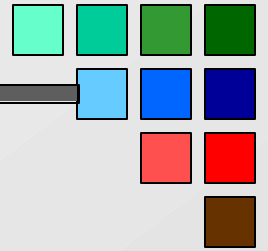
(b)

# Replicação de Objetos (3)

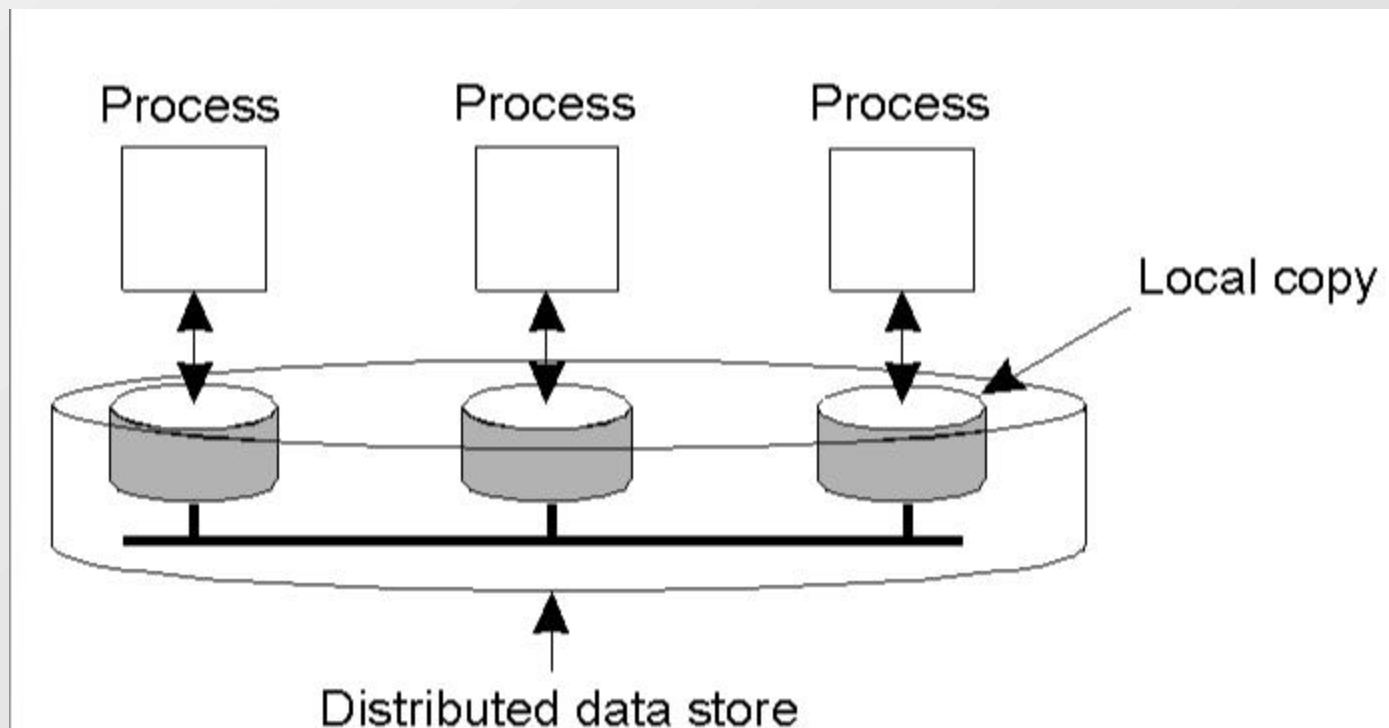


- Um SD para replicação de objetos distribuídos
- Um SD responsável por gerenciamento de réplicas

# Modelos de consistência voltados para dados

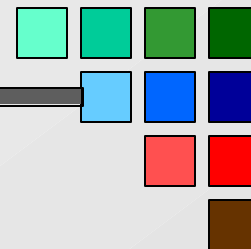


- A organização geral de um conjunto de dados, fisicamente distribuídos e replicados através de múltiplos processos





# Consistência restrita



P1:  $W(x)a$   
P2:  $R(x)a$   
(a)

P1:  $W(x)a$   
P2:  $R(x)NIL$   $R(x)a$   
(b)

- Comportamento de dois processos operando sobre os mesmos dados
- (a) Um conjunto de dados operando sobre consistência restrita
- (b) Um conjunto de dados que não possui consistência restrita

# Consistência seqüencial e linearidade

(1)

P1:	W(x)a		
P2:	W(x)b		
P3:		R(x)b	R(x)a
P4:		R(x)b	R(x)a

(a)

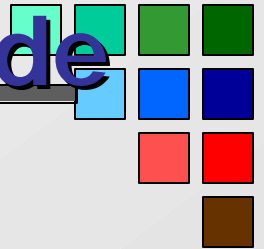
P1:	W(x)a		
P2:	W(x)b		
P3:		R(x)b	R(x)a
P4:		R(x)a	R(x)b

(b)

- Um conjunto de dados consistente
- Um conjunto de dados que não é consistente seqüencialmente

# Consistência seqüencial e linearidade

## (2)



### Process P1

---

```
x = 1;  
print ( y, z);
```

### Process P2

---

```
y = 1;  
print (x, z);
```

### Process P3

---

```
z = 1;  
print (x, y);
```

- Três processos executando concorrentemente

# Consistência seqüencial e linearidade

(3)

- Quatro seqüências válidas de execução para os processos do slide anterior

```
x = 1;  
print (y, z);  
y = 1;  
print (x, z);  
z = 1;  
print (x, y);
```

Prints: 001011

Signature:  
001011

(a)

```
x = 1;  
y = 1;  
print (x,z);  
print(y, z);  
z = 1;  
print (x, y);
```

Prints: 101011

Signature:  
101011

(b)

```
y = 1;  
z = 1;  
print (x, y);  
print (x, z);  
x = 1;  
print (y, z);
```

Prints: 010111

Signature:  
110101

(c)

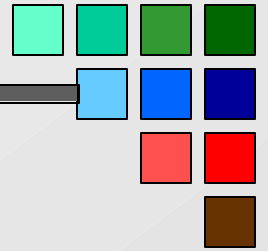
```
y = 1;  
x = 1;  
z = 1;  
print (x, z);  
print (y, z);  
print (x, y);
```

Prints: 111111

Signature:  
111111

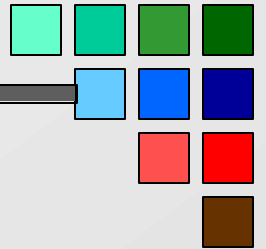
(d)

# Consistência Causal (1)



- Condição necessária:  
Escritas que são relacionadas por **causa** devem ser vistas por todos os processos na mesma ordem. Escritas concorrentes podem ser vistas em uma ordem diferente em diferentes máquinas

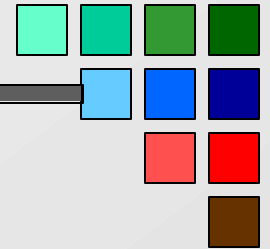
# Consistência Causal (2)



P1:	$W(x)a$		$W(x)c$	
P2:		$R(x)a$	$W(x)b$	
P3:		$R(x)a$		$R(x)c$
P4:		$R(x)a$		$R(x)b$

- Essa seqüência é permitida com um conjunto de consistência causal mas não com conjuntos seqüenciais e de consistência restrita

# Consistência Causal (3)



P1:	W(x)a		
P2:	R(x)a	W(x)b	
P3:			R(x)b R(x)a
P4:			R(x)a R(x)b

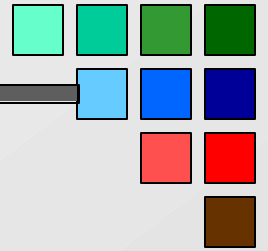
(a)

P1:	W(x)a		
P2:	W(x)b		
P3:			R(x)b R(x)a
P4:			R(x)a R(x)b

(b)

- Uma violação de um conjunto de consistência causal
- Uma seqüência correta (pois não há relação de causa) de eventos em um conjunto de consistência causal

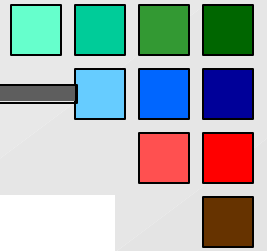
# Consistência FIFO (1)



- Condição Necessária:
  - Escritas feita por um único processo são vistas por todos os outros na ordem em que elas são emitidas
  - Escritas de diferentes processos podem ser vistas em uma ordem diferente por diferente processos



# Consistência FIFO (2)



P1: W(x)a

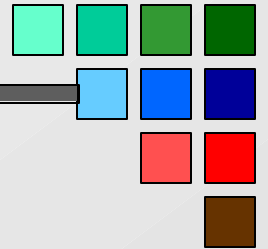
P2: R(x)a W(x)b W(x)c

P3: R(x)b R(x)a R(x)c

P4: R(x)a R(x)b R(x)c

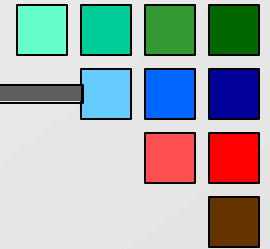
- Uma seqüência válida de eventos FIFO

# Consistência Fraca (1)



- Propriedades:
  - Acessos para variáveis de sincronização associados com um conjunto de dados são seqüencialmente consistentes
  - Nenhuma operação na variável de sincronização é permitida até que todas as escritas anteriores tenham sido completadas
  - Nenhuma operação de leitura ou escrita nos itens de dados são permitidos até que todas as operações anteriores nas variáveis de sincronização tenham sido realizadas

# Consistência fraca (3)



P1:	W(x)a	W(x)b	S			
P2:				R(x)a	R(x)b	S
P3:				R(x)b	R(x)a	S

(a)

P1:	W(x)a	W(x)b	S			
P2:				S	R(x)a	

(b)

- Uma seqüência válida de eventos para consistência fraca
- Uma seqüência válida para consistência fraca

# Consistência de Atualização (1)

P1:	Acq(L)	W(x)a	W(x)b	Rel(L)	
P2:			Acq(L)	R(x)b	Rel(L)
P3:					R(x)a

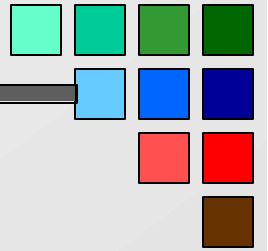
- Uma seqüência de eventos válida para consistência de atualização

# Consistência de Atualização (2)

- Regras:

- Antes que uma operação de leitura ou escrita sobre dados compartilhados seja realizada, todas as aquisições feitas por processos devem ter sido completadas com sucesso
- Antes que uma atualização seja permitida para ser realizada, todas as leituras e escritas anteriores realizadas pelos processos, devem ter sido completadas
- Acessos para variáveis de sincronização são de consistência FIFO (consistência seqüencial não é necessária)

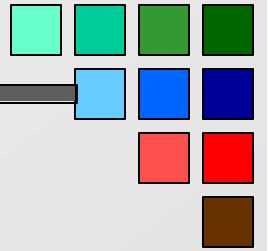
# Consistência de Entrada (1)



- Condições:

- Uma aquisição de acesso de uma variável de sincronização não pode ser realizada sobre um processo até que todas as atualizações para os dados compartilhados guardados, tenham sido realizados sobre esse processo
- Antes que um modo de acesso exclusivo para uma variável de sincronização por um processo é permitido para ser realizado com relação a esse processo, nenhum outro processo pode manter a variável de sincronização, nem mesmo em modo exclusivo
- Depois que um modo de acesso exclusivo para uma variável de sincronização seja processado, qualquer outro modo de acesso não-exclusivo do processo para essa variável de sincronização pode não ser realizado até que tenha realizado para essa própria variável

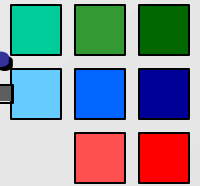
# Consistência de Entrada (1)



P1:	Acq(Lx)	W(x)a	Acq(Ly)	W(y)b	Rel(Lx)	Rel(Ly)
P2:					Acq(Lx)	R(x)a
P3:						R(y)b

- Uma seqüência de eventos válida para uma consistência de entrada

# Resumo dos Mod. de Consist.



Consistência	Descrição
Restrita	Ordem de tempo absoluta de todos os acessos compartilhados
Linearidade	Todos os processos devem ver todos os acessos compartilhados na mesma ordem. Acessos são ordenados de acordo com um timestamp global
Seqüencial	Todos os processos veem acessos compartilhados na mesma ordem. Acessos não são ordenados no momento
Causal	Todos os processos veem acessos compartilhados relacionados por causalidade na mesma ordem
FIFO	Todos os processos veem escritas dos outros na ordem em que elas são usadas. Escritas de diferentes processos podem nem sempre estar nessa mesma ordem

(a)

Consistência	Descrição
Fraca	Dados compartilhados podem ser contados para ser consistente apenas depois que uma sincronização é feita
Atualização	Dados compartilhados são consistentes quando uma região crítica é deixada
Entrada	Dados compartilhados que pertencem para uma região crítica são consistentes quando entra-se em uma região crítica

(b)

- Modelos de consistência que não usam operações de sincronização
- Modelos com operações de sincronização