


© 2004-2015 Volnys Bernal 1

Semáforo

Volnys Borges Bernal
volnys@lsi.usp.br

Departamento de Sistemas Eletrônicos
Escola Politécnica da USP




© 2004-2015 Volnys Bernal 2

Sumário

- ❑ Mecanismos de sincronização e comunicação:
 - ❖ Semáforo
 - Semáforo
 - Semáforo binário
 - Interface pthreads para semáforo

© 2004-2015 Volnys Bernal 3

Semáforo



© 2004-2015 Volnys Bernal 4

Semáforo

- ❑ Método de sincronização que permite a contagem de recursos disponíveis
- ❑ Primitivas
 - ❖ Up(semáforo)
 - Incrementa o contador do semáforo.
 - Se existirem entidades bloqueadas neste semáforo, uma delas é desbloqueada
 - ❖ Down(semáforo)
 - Decrementa o semáforo
 - Se o resultado for menor que zero, a entidade fica bloqueada neste semáforo.
 - ❖ Init(semáforo, valor)
- ❑ Estas primitivas são garantidamente atômicas (indivisíveis)
- ❑ O semáforo deve ser iniciado com um valor inteiro.
- ❑ Normalmente, este valor está associado à quantidade de recursos disponíveis.

© 2004-2015 Volnys Bernal 5

Semáforo

- ❑ Uso de semáforo no problema do produtor-consumidor
 - ❖ Necessita 3 semáforos
 - 1 para garantir a exclusão mútua
 - 1 para bloquear os produtores
 - quando a fila estiver cheia (sem slots disponíveis)
 - 1 para bloquear os consumidores
 - quando a fila estiver vazia (sem itens na fila)

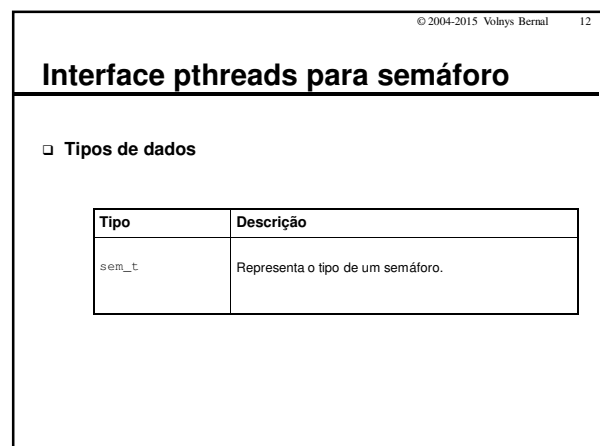
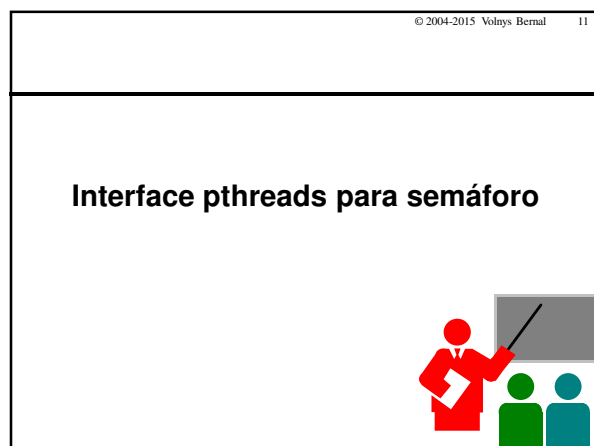
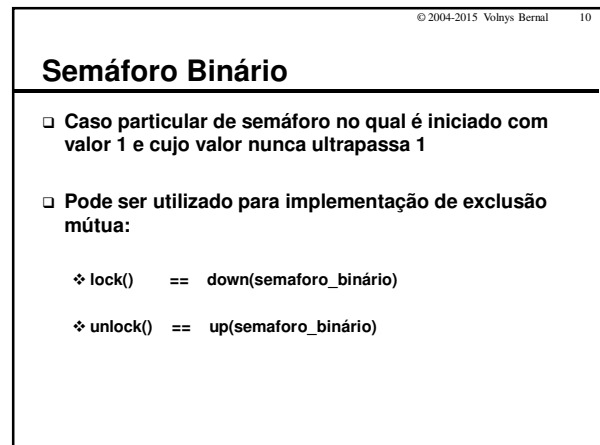
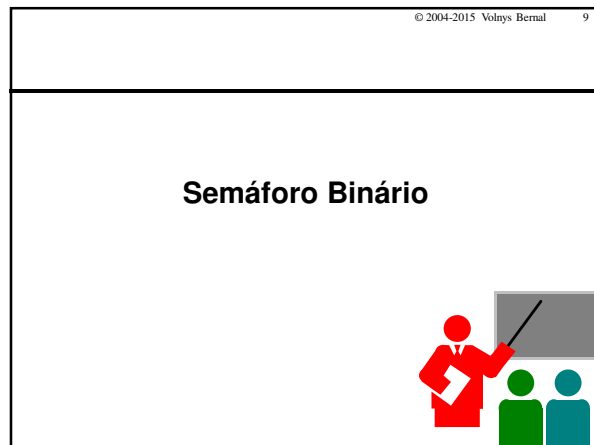
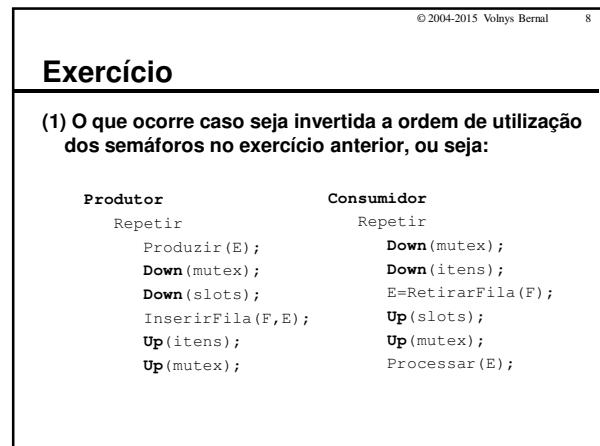
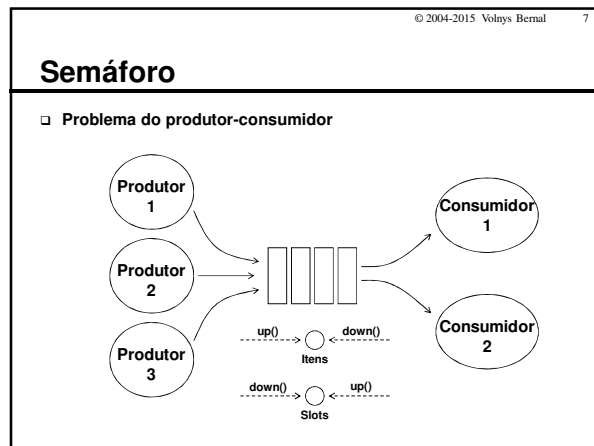
© 2004-2015 Volnys Bernal 6

Semáforo

- ❑ Problema do produtor-consumidor

```
semaforo mutex = 1;
semaforo slots = N;
semaforo itens = 0;
```

Produtor	Consumidor
Repetir	Repetir
Produzir(E);	Down(itens);
Down(slots);	Down(mutex);
Down(mutex);	E=RetirarFila(F);
InserirFila(F,E);	Up(mutex);
Up(mutex);	Up(slots);
Up(itens);	Processar(E);



© 2004-2015 Volnys Bernal 13

Interface pthreads para semáforo

❑ Primitivas

Primitiva	Descrição
<code>sem_init</code>	Iniciação da variável de um semáforo.
<code>sem_wait</code>	Down. Decrementa o semáforo. Se o valor resultante for menor que zero a entidade de processamento é bloqueada.
<code>sem_trywait</code>	Variante de Down. Caso o valor do semáforo seja igual ou menor que zero, retorna. Caso contrário, decrementa o semáforo.
<code>sem_post</code>	Up. Incrementa o semáforo. Se existirem entidades de processamento bloqueadas neste semáforo, uma delas é desbloqueada.
<code>sem_get_value</code>	Retorna o contador do semáforo.
<code>sem_destroy</code>	Destroi um semáforo.

© 2004-2015 Volnys Bernal 14

Interface pthreads para semáforo

❑ Sintaxe das primitivas

```
#include <semaphore.h>

int sem_init (sem_t *sem, int pshared, unsigned int value)
int sem_wait (sem_t *sem)
int sem_trywait (sem_t *sem)
int sem_post (sem_t *sem)
int sem_getvalue (sem_t *sem, int *sval)
int sem_destroy (sem_t *sem)
```

© 2004-2015 Volnys Bernal 15

Interface pthreads para semáforo


❑ Exemplo de uso

```
#include <semaphore.h>

...
sem_t slots;
...
status = sem_init(&slots,0,10);
...
status = sem_wait(&slots);
...
status = sem_post(&slots);
...
```

© 2004-2015 Volnys Bernal 16

Referências Bibliográficas



© 2004-2015 Volnys Bernal 17

Referências Bibliográficas

- ❑ ANDREW S. TANENBAUM; Sistemas Operacionais Modernos. Prentice-Hall.
❖ Capítulo 2
- ❑ ANDREW S. TANENBAUM; Sistemas Operacionais. Prentice-Hall.
❖ Capítulo 2