

# Sistemas Distribuídos

## Introdução: Comunicação, Falhas e Coordenação

março de 2015

# Comunicação entre Processos

- troca de mensagens é sempre a primitiva básica

- sobre essa primitiva podemos construir outras visões da comunicação

# Mensagens Básicas

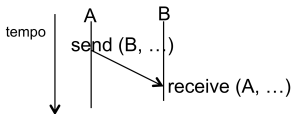
```
send (destino, &mensagem)
```

```
receive (origem, &mensagem)
```

- questões
  - semântica de operações
  - especificação de origem e destino
  - formato da mensagem

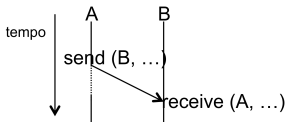
## semântica — envio síncrono e assíncrono

- envio assíncrono: execução procede imediatamente
  - bufferização

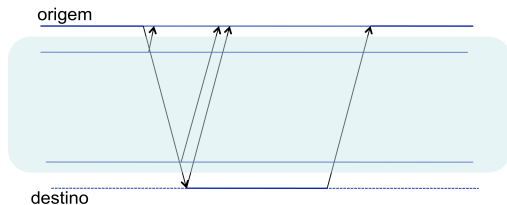


## semântica — envio síncrono e assíncrono

- envio síncrono: execução procede quando destinatário recebe mensagem
  - determinismo



## Variantes síncronas e assíncronas



algumas bibliotecas oferecem primitivas com várias diferentes semânticas de garantia de envio

## recebimento síncrono e assíncrono

- recebimento síncrono é o convencional
  - execução procede quando há algo a tratar
  - alternativa de recebimento com timeout

### bloqueio!

suspensão da linha de execução corrente até chegada da mensagem

## bloqueio e recebimento explícito

- espera por mensagens de tipos específicos?
  - podem haver outros tipos de mensagens pendentes
- espera por qualquer tipo de mensagem
  - código se torna um enorme emaranhado



## recebimento assíncrono

- orientação a eventos: chegada de mensagem é encarada como evento a ser tratado
  - recebimento fica implícito
- o *recebimento implícito* é considerado um recebimento sem bloqueio

outras formas de recebimento sem bloqueio serão discutidas mais tarde

## Especificação de destino

- endereçamento direto (acoplamento)
  - endereços e volatilidade
  - serviços de nomes
- caixas de correios e canais (desacoplamento)
  - endereços bem conhecidos
  - comunicação com par “qualquer”

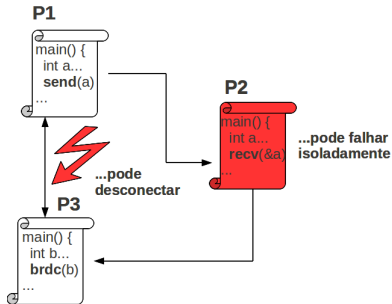
## Formato de mensagens

- no nível mais baixo: sequências de bytes
  - interpretação por conta do programa

## Formato de mensagens

- apoio de bibliotecas ou linguagens a empacotamento e desempacotamento
  - marshalling e unmarshalling
- abstrações em vários níveis

## Falhas parciais



A noção de “falha parcial” é fundamental nos SDs: a cooperação entre processos deve ser *tolerante* a falhas individuais

## Falhas – Nomenclatura

- problemas inevitáveis: falhas (faults)
  - máquinas quebradas, desconexões, erros no software
- erros (failures): consequências dessas falhas

nomenclaturas variam mas temos que manter consistente a idéia de evitar que falhas se transformem em erros

## Falhas – Nomenclatura

- omissão
- temporização
- falhas arbitrárias ou bizantinas

# Dependabilidade

- termo clássico “tolerância a falhas” por vezes considerado enganador
  - dependabilidade: ideia de que se pode confiar no sistema (apesar de possíveis erros)

## dependabilidade

- reliability
- availability
- safety
- security



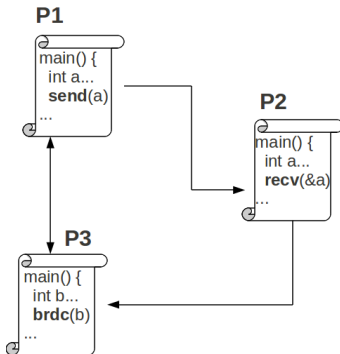
## Processos e enlaces

Usaremos duas abstrações principais para representar o sistema físico: *processos* e *enlaces*

- *Processos*: abstraem *entidades ativas* e executam processamento (pode representar um *computador*, um *processador*, uma *thread*, etc.)
- *Enlace*: abstraem a *rede de comunicação* física lógica que permite a troca de mensagens entre processos

# Coordenação

- processos em aplicação distribuída devem se *coordenar* para que aplicação progrida



## Coordenação e Padrões

utilização de *padrões* recorrentes de interação

muitos problemas recaem em versões diversas do *problema de acordo distribuído*: ex., *processos concordam sobre a ordem de tratamento de mensagens, se um evento ocorreu ou não, etc.*

...assunto específico do curso de algoritmos distribuídos

## Coordenação: acordo sobre hora

- hora certa é conceito extremamente útil para sincronizar atividades no “mundo real”
- em sistemas computacionais:
  - ordenação
  - depuração
  - escalonamento de atividades
  - medidas
  - ...
- necessidade de uma hora certa *global*



## Sincronização de relógios físicos

### interna

- diferença máxima entre horas de dois relógios do sistema

### externa

- diferença máxima entre horas de qualquer relógio do sistema e uma fonte externa

## Algoritmos de sincronização de relógio

- trocas de mensagens repetidas para obtenção de diversas amostras
- protocolo *NTP* atualmente o mais difundido

