Aula 17 de FSO

José A. Cardoso e Cunha
DI-FCT/UNL

Este texto resume o conteúdo da aula teórica.

1 Objectivo

Objectivo da aula: Comunicação por mensagens. Comunicação síncrona e assíncrona. Fluxo da informação entre processos. Interacção cliente-servidor.

2 Sincronização das operações de envio e de recepção

2.1 Envio de mensagens

Podemos ter o envio não bloqueante ou assíncrono: o emissor não espera por qualquer confirmação de que a mensagem tenha sido recebida.

O envio pode ser bloqueante só até o SO confirmar que copiou a mensagem para zonas de buffers próprias e iniciou o envio. Ou pode ser bloqueante até que o emissor receba uma confirmação de que a mensagem foi efectivamente recebida. Neste segundo caso, o envio também se diz síncrono, pois a comunicação processa-se como se tanto o emissor como o receptor aguardassem um pelo outro, até se poder transmitir a mensagem, só prosseguindo depois disso. No modelo síncrono de comunicação, os processos emissor e receptor evoluem 'passo a passo', trocando sinais de controlo (confirmação de recepção), sempre que há mensagens trocadas entre eles.

No modelo assíncrono, os processos evoluem aos seus próprios ritmos, recorrendo ao SO para armazenar temporariamente, em buffers seus, as mensagens pendentes para serem entregues. Os emissores não têm confirmação de que as suas mensagens sejam recebidas. Este modelo permite maior paralelismo nas acções, pois não condiciona os emissores a esperarem pelos receptores, mas torna mais difícil a compreensão da evolução das computações, ao contrário do modelo síncrono.

2.2 Recepção de mensagens

A recepção pode ser:

- bloqueante: o receptor bloqueia-se até que lhe seja entregue uma mensagem (é este o modo mais habitual, por exemplo, no Unix System V, é o modo pré-definido (*WAIT*) na chamada ao SO *msgrcv()*
- não bloqueante: tem duas possibilidades:
  - a operação *receber* devolve um erro indicando que não há mensagens pendentes, mas retorna de imediato ao processo invocador;
  - o processo receptor declarou um procedimento (*handler*) de tratamento que será invocado quando chegar alguma mensagem (notificação assíncrona).

3 Fluxo da informação entre processos

3.1 Unidireccional

Tal como sucede nos *pipes* no Unix, o fluxo da informação trocada entre processos, em cada primitiva de comunicação, é, na sua forma mais básica, unidireccional, do emissor para o receptor (figura 1).

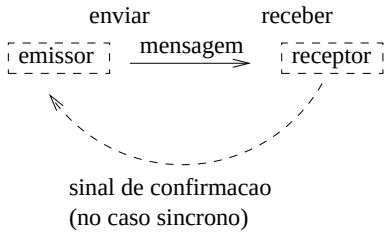

Figura 1: Fluxo unidireccional.

3.2 Bidireccional

Modos mais sofisticados podem ser realizados com base no fluxo unidireccional, conforme se ilustra na figura 2).

No caso (a), trata-se de uma simples troca de duas mensagens, num mesmo acto de comunicação, em geral associado a um ponto de encontro

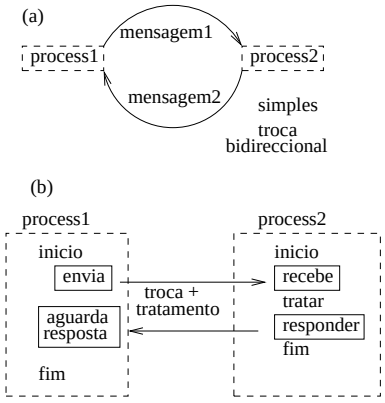

Figura 2: Fluxo bidireccional.

(*rendez-vous*) dos dois processos envolvidos, isto é, o primeiro a chegar bloqueia-se até que o segundo chegue e, então, trocam as mensagens. É um mecanismo muito expressivo que surge em diversos modelos e linguagens de programação concorrente. Pode ser implementado com base no modelo básico de comunicação unidireccional, pelo que não é suportado habitualmente a nível das chamadas ao SO.

No caso (b), há também uma troca de duas mensagens, uma em cada sentido. Do ponto de vista do processo 1 (na figura 2), o mecanismo é o mesmo que no caso (a), mas o mesmo não se passa quanto ao comportamento do processo 2 (na figura 2), pois que este processo, uma vez recebida a mensagem do processo 1, efectua um processamento local e só depois responde, com o envio de uma mensagem ao processo 1. Poderíamos chamar a esta interacção uma 'troca retardada' de mensagens, e ao caso (a), uma 'troca imediata', mas a verdade é que o interesse do caso (b) é o de permitir modelar as interacções típicas entre processos clientes e servidores. Estas interacções, se programadas no modelo básico unidireccional, têm a forma:

```
processo cliente          processo servidor

enviar(...);              receber(...);
receber(...);             enviar(...);
```

No caso do modelo bidireccional (b), tem-se directamente representada a interacção do cliente (processo 1) com o servidor (processo 2).

4 Interacções simétricas ou assimétricas

Esta dimensão descreve o papel dos participantes na comunicação. No caso da comunicação bidireccional 'imediata' (modo (a) da figura 2), os dois processo tem papéis simétricos, ambos enviando e recebendo. No caso (b), contudo, o cliente tem um papel de invocador, que pede um 'serviço', e o servidor tem um papel de receber pedidos, que trata. O servidor pode ter um papel passivo, isto é, aceita todos os pedidos incondicionalmente, por exemplo, numa fila única de mensagens, após o que trata um pedido de cada vez, ou pode, em alternativa, decidir, a cada momento, quais os pedidos que aceita e os que recusa.

Existem diversas formas de um servidor seleccionar, filtrar ou discriminar entre múltiplas classes de pedidos:

- receber apenas mensagens de um certo tipo (ver, por exemplo, o caso Unix System V);
- abrir / fechar portas de comunicação onde recebe os pedidos;
- invocar uma primitiva de recepção de mensagens apenas sobre um determinado conjunto de portas ou caixas de correio.

Voltaremos a este assunto quando estudarmos os mecanismos mais estruturados de comunicação, nas aulas seguintes.