
Programação de Sistemas

Introdução aos modelos de comunicação

Introdução (1)

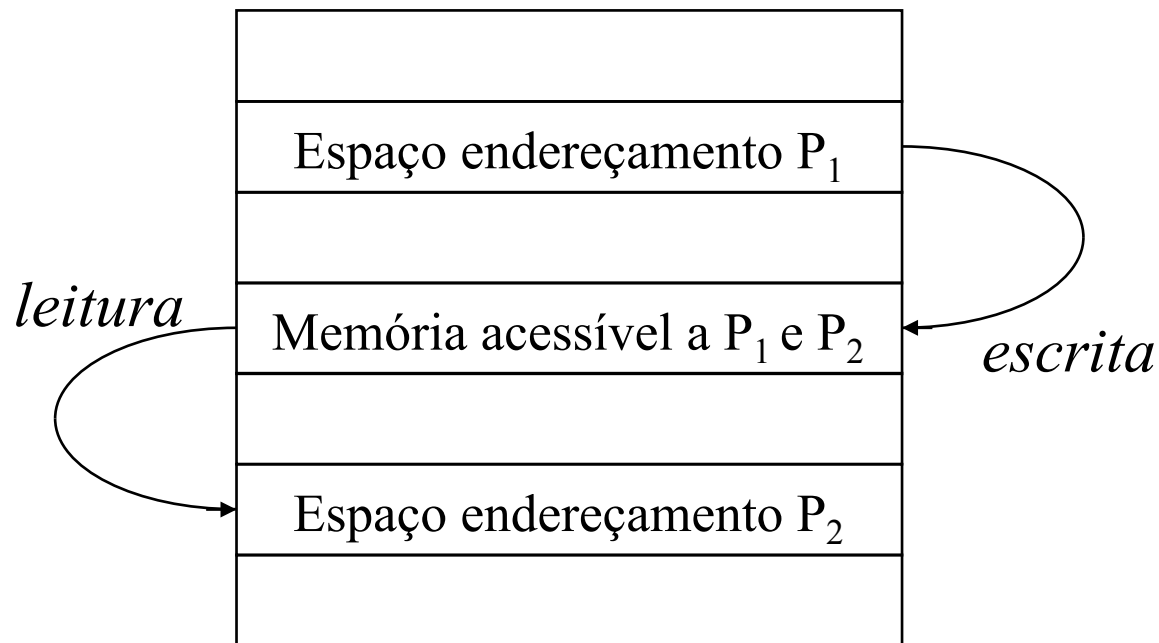
- Os sistemas operativos implementam mecanismos que asseguram independência entre processos (i.e., a execução de um processo não afecta os outros, para além da variação do tempo de duração).
- No entanto, em muitos casos os processos necessitam trocar informação de forma controlada para
 - dividir tarefas,
 - aumentar a capacidade de processamento por distribuição do esforço em computadores distintos ligados em rede,
 - sincronizar a execução de cada um.

Introdução (2)

- Os mecanismos de comunicação são classificados em
 - A. Comunicação por memória partilhada
 - Semáforos, mutexes (estudados no capítulo de sincronização de processos)
 - Segmentos partilhados (**Nota**: não são abordados nesta disciplina)
 - B. Comunicação via núcleo, que exige cópia de dados. Os mecanismos disponíveis dependem da arquitectura do sistema:
 - Ambiente centralizado
 - Fluxo de Bytes (Pipes)
 - Sinais
 - Ambiente distribuído (ex: redes de computadores)
 - Mensagens (Sockets)
 - Chamada remota, com dados passados por parâmetro. O mecanismo mais divulgado é o RPC – “Remote Procedure Call” (**Nota**: não é abordado nesta disciplina).

Introdução (3)

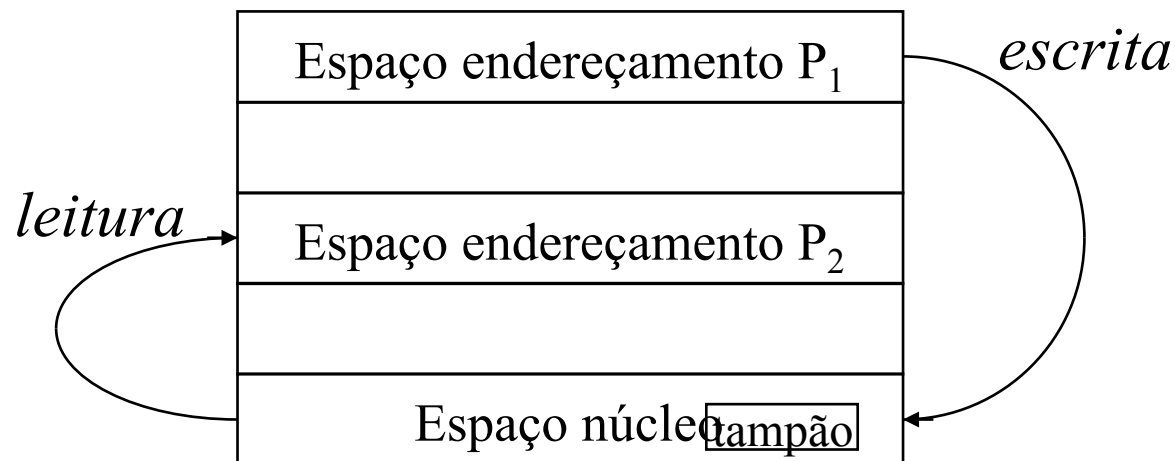
A. Comunicação via memória partilhada



- Vantagens da comunicação via memória partilhada: mais eficiente (cópia de dados consome recursos)
- Inconvenientes: problemas de sincronização

Introdução (4)

B. Comunicação via núcleo



- Vantagens da comunicação via núcleo
 - Pode ser realizada em sistemas com vários CPU.
 - Sincronização implícita.

Introdução (5)

- Comunicação via núcleo é executada na seguinte sequência de passos:
 1. Criação de um canal do processo produtor para o processo consumidor, via núcleo.

Nota: o canal pode ser interno ao computador, ou ligando dois computadores.
 2. Processo produtor escreve mensagem no canal.

Nota: a escrita é não bloqueante (admitindo que o núcleo tem uma memória tampão com capacidade de armazenar todas as mensagens)
 3. Processo consumidor lê mensagem vinda do canal.

Nota: a leitura é bloqueante, o consumidor fica suspenso até à chegada da mensagem.

Introdução (6)

[Definição] **IPC**-”InterProcess Communication” é o conjunto de mecanismos de troca de informação entre múltiplas threads de um ou mais processos.

- O Linux disponibiliza vários tipos de mecanismos IPC, de acordo com os objectivos de cooperação:
 - E/S de uso geral: tubos (“*pipes*”) / filas (“*fifos*”), STREAMS.
 - E/S entre computadores distintos: *Sockets* e RPC-”Remote Procedure Call”.
 - Controlo de processos: *signal*.
 - Protecção de acesso a recursos (sincronização de processos): semáforos e *mutexes*.

Introdução (7)

[Def] Canal de comunicação : meio pelo qual a informação é transmitida do emissor ao receptor.

Exemplos:

- Memória partilhada pelos processos emissor e receptor.
- Memória tampão no núcleo.
- Nível de transporte na Internet, responsável pela transferência do nó fonte ao nó destino independentemente da forma como a rede de computadores subjacente se encontra organizada.

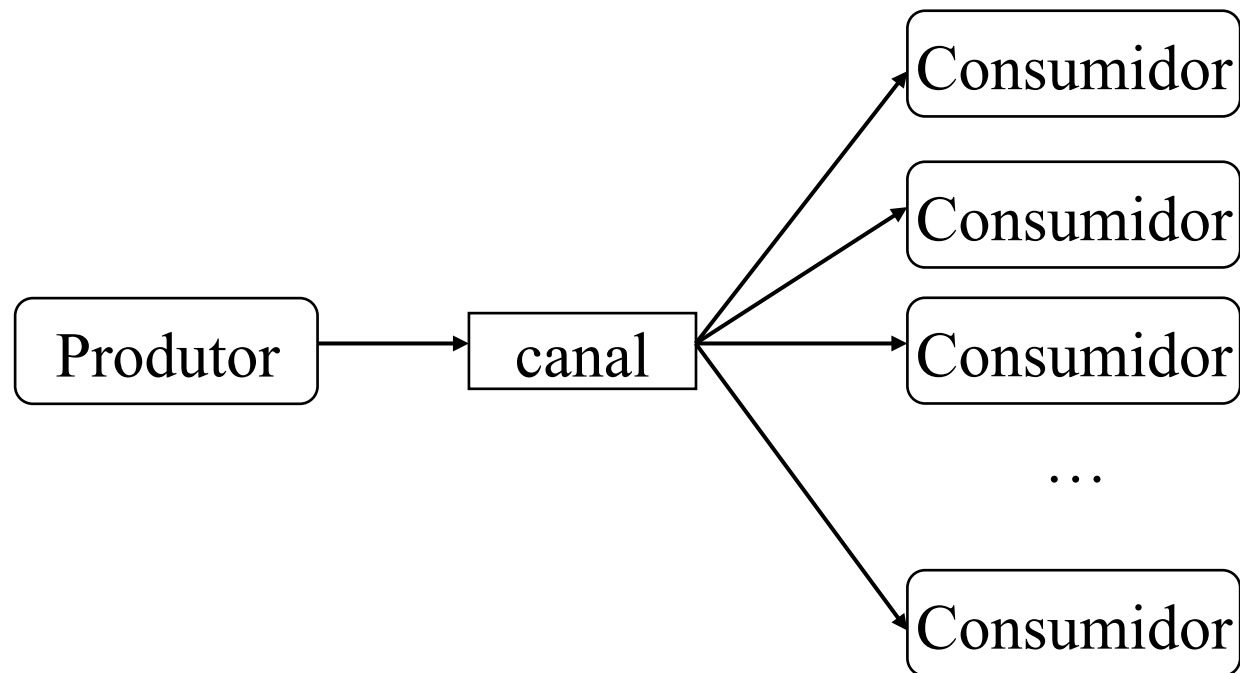
[Def] Protocolo : regras de troca de mensagens entre os intervenientes.

Modelos de comunicação (1)

- A arquitectura de comunicação entre diversos intervenientes (programas ou processos) segue um modelo determinado. Modelos de comunicação entre aplicações
 - Difusão (“*broadcast*”) : o emissor envia a mesma informação a todos os restantes processos receptores.
 - Produtor-consumidor: comunicação uni-direccional, com o produtor a entregar mensagens ao consumidor.
 - Cliente-servidor : cliente controla totalmente o servidor.
 - Caixa de correio : as mensagens podem ser lidas por um processo receptor sem o emissor (um entre vários) poder controlar o processo que recolhe a mensagem.
 - Diálogo : dois processos acordam canal temporário para troca de mensagens durante uma sessão.

Nota: matéria central na disciplina
Redes de Computadores

Modelo de difusão



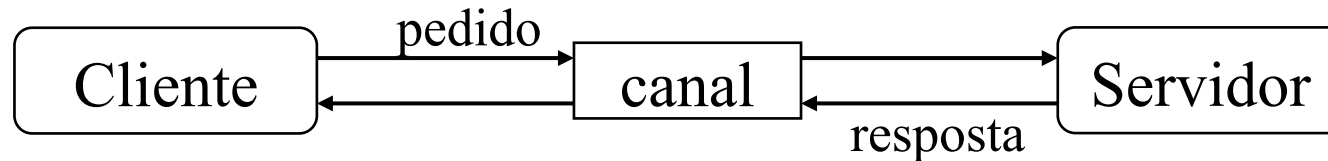
- O produtor envia mensagem a todos os consumidores, sem saber quem e quantos são.
- “*Broadcast*” em Inglês

Modelo produtor-consumidor



- Conexão uni-direccional fixa, do produtor para o consumidor.
- Exemplo de implementação analisado num exercício teorico-pratico no capítulo de sincronização de processos.

Modelo cliente-servidor (1)



- Conexão bi-direccional fixa, entre o cliente (computador, programa ou processo que requiere informação) e o servidor (computador, programa ou processo que disponibiliza determinado serviço ao cliente).
- A comunicação desenrola-se nos seguintes passos:
 1. O cliente envia um pedido ao servidor.
 2. O servidor executa o pedido.
 3. O servidor assinala o resultado ao cliente.

Nota: normalmente cliente e servidor residem em computadores distintos (ex: WWW), mas pode haver exceções (ex: sistema de janelas X).

Modelo cliente-servidor (2)

[Def] Programação modular : software de grande dimensão é dividido em módulos, facilitando o desenvolvimento e a manutenção.

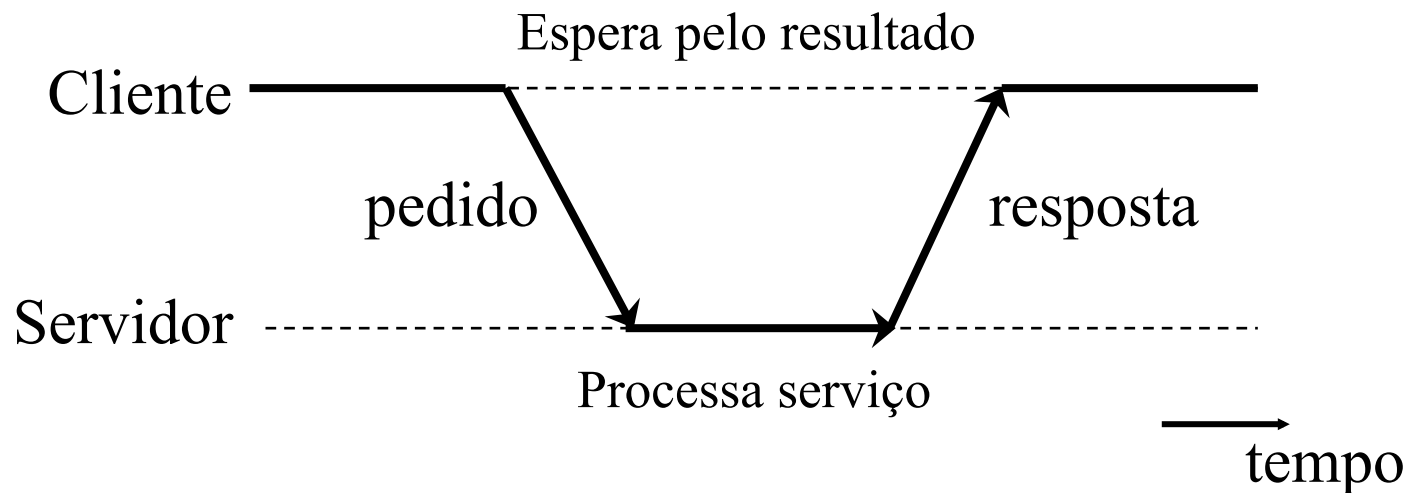
- O modelo cliente-servidor estende o conceito de programação modular, na medida que os módulos podem ser executados em espaços distintos de memória.

[Def] Cliente : módulo chamador, que requiere serviço.

[Def] Servidor : módulo chamado, que disponibiliza serviço.

Modelo cliente-servidor (3)

- Tipicamente, a interacção entre o cliente e o servidor segue as seguintes etapas



Nota: transferência de um ficheiro designada por “*download*” (do servidor para o cliente) ou “*upload*” (do cliente para o servidor).

Modelo cliente-servidor (4)

- O modelo cliente-servidor tem sido adoptado em muitos serviços da Internet.

Exemplos de protocolos definidos no modelo de comunicação cliente-servidor:

- HTTP para WWW.
- SMTP para transferência de Email entre nós.
- POP3 para utilizador recolher mensagens Email guardadas num ISP-Internet Service Provider.

Modelo cliente-servidor (5)

- Fixadas as normas dos protocolos de comunicação entre cliente e servidor, cada um pode ser implementado de forma independente.

Exemplos:

- Servidores WWW
 - Apache (<http://httpd.apache.org>)
 - lighttpd (<http://www.lighttpd.net>)
- Clientes WWW (navegadores-"browsers")
 - Mosaic, primeiro navegador gráfico.
 - Firefox (<http://www.mozilla.com/>), derivado do Mozilla, corre em várias plataformas (Linux, Mac OS, Windows).
 - Internet Explorer, actualmente em actualização apenas para Windows.
 - Lynx (<http://lynx.browser.org/>), para terminais textuais.

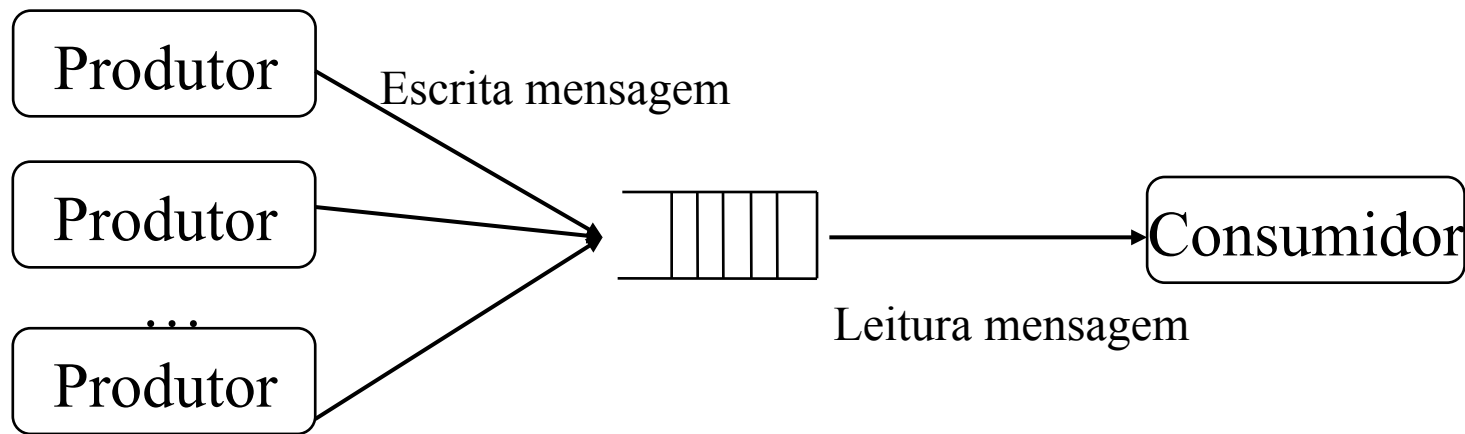
Modelo P2P

- No modelo cliente-servidor as tarefas são separadas por um dos dois nós intervenientes.
Os dados são divididos numa partição.
- No modelo P2P-”peer to peer”, o nó é simultaneamente cliente e servidor.
Os dados são divididos aleatoriamente pela rede: dados localizados por “inundação” da rede com pedidos.

Exemplos:

- Rede de notícias *Usenet* (1980)-em 2007 circulavam diariamente 3TB, uma parte significativa SPAM.
- *Kazaa* (1999), que possui nós dedicados para fornecer localização de dados. Processado por empresas discográficas, faliu em 2002.

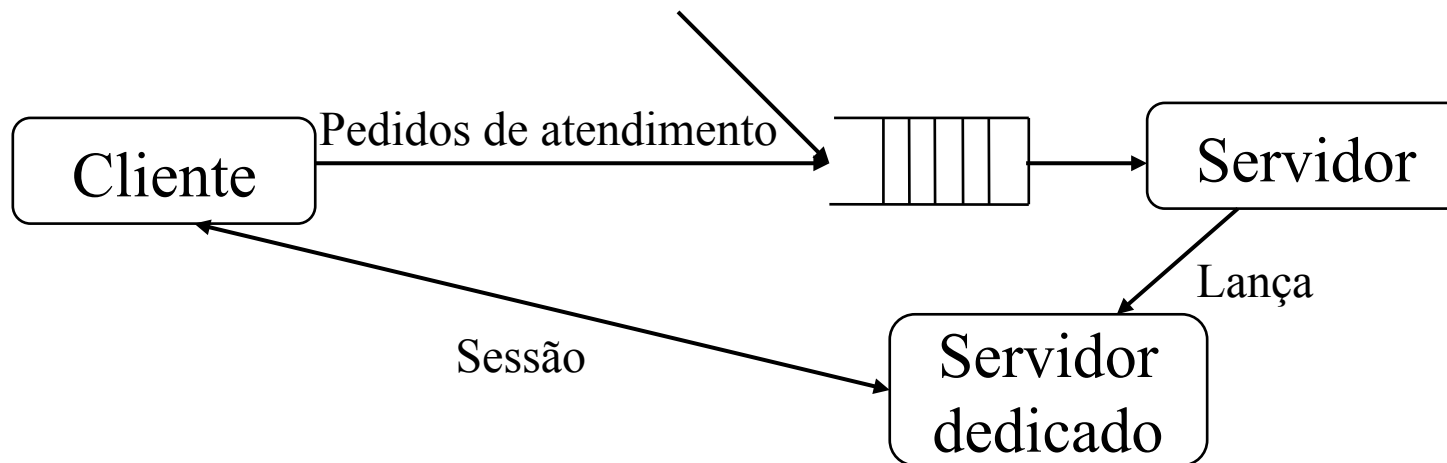
Modelo caixa de correio



- A ligação entre produtor e consumidor é indirecta.
 1. Produtor escreve mensagem na caixa do correio.
 2. O consumidor recolhe uma mensagem da caixa de correio, sem escolha do produtor que escreveu a mensagem.

Nota: Tal como no modelo produtor-consumidor, a escrita não é bloqueante (admitindo que a capacidade da caixa de correio é ilimitada) e a leitura é bloqueante quando a caixa se encontra vazia.

Modelo diálogo



1. O cliente envia um pedido ao servidor.
2. Criado um servidor dedicado, que liga ao cliente por um canal dedicado.
3. O canal   desligado quando a sess o termina.