

Sistemas operacionais: Conceitos e Mecanismos

Capítulo 9: Mecanismos de comunicação

Integrantes do grupo: Jefferson Botitano e Leonardo Ludvig
Professor: Arliones Stevert Hoeller Junior
Disciplina: Sistemas operacionais

Pipes

Os pipes são um dos mecanismos de comunicação mais simples utilizados pelos sistemas operacionais sendo um canal de comunicação local unidirecional entre dois processos, são considerados canais de comunicação confiáveis, orientado a fluxo e com capacidade finita, visto que fazem a comunicação por meio da memória do núcleo para interligar os dois processos. Usualmente os pipes padrão tem uma vida curta pois são destruídos logo após a execução do processo ou comando que os criou, porém também existem os “named pipes” ou “FIFOs” os quais permanecem desde a sua criação até o sistema ser encerrado ou serem explicitamente destruídos, esses pipes com nomes próprio são independentes de processos e podem ser encontrados por qualquer processo através desse nome, podendo ser criados por linha de comandos ou programas.

Filas de mensagens

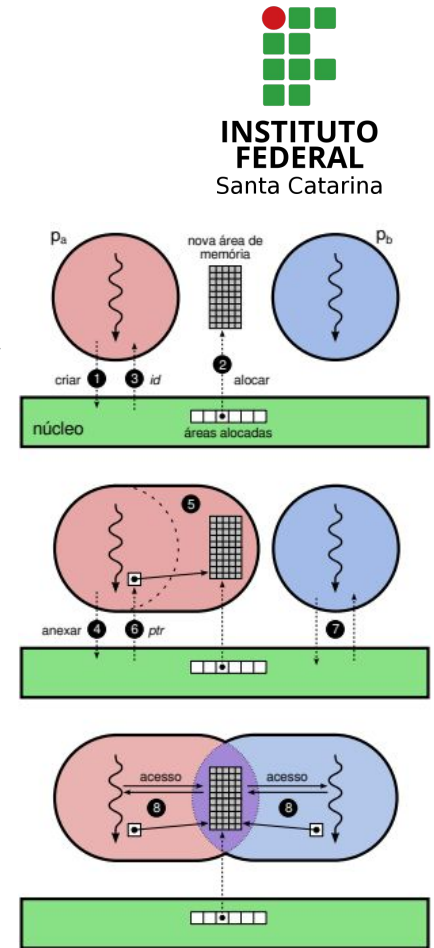
Para possibilitar uma comunicação com envio e recebimento de dados tipados entre vários processos, podendo ter a abordagem N:1 ou N:M dependendo da sua implementação em um sistema operacional, foi criado o conceito de filas, as quais são canais de comunicação confiáveis, orientadas a mensagens e com capacidade finita, visto que também utilizam da memória do núcleo para realizar a transmissão de dados entre os processos. As operações de envio e recepção de uma fila podem variar de síncronas ou assíncronas dependendo do critério abordado para sua criação, alguns exemplos de chamadas para usar filas são:

- **mq_open:** Cria ou abre uma fila já existente.
- **mq_setattr e mq_getattr:** Permite ajustar ou obter parâmetros de uma fila.
- **mq_send:** Envia uma mensagem a fila e caso a fila caso esteja cheia bloqueia o emissor até abrir espaço.
- **mq_receive:** Recebe uma mensagem da fila, caso a fila caso esteja vazia bloqueia o receptor até surgir uma mensagem para ser recebida..
- **mq_close:** fecha o descritor da fila criado por mq_open
- **mq_unlink:** remove a fila do sistema, destruindo seu conteúdo.

Memória compartilhada

Em comunicações onde tarefas estão situadas em processos distintos devem ser realizadas através do núcleo por chamadas de sistemas, pois não existe a possibilidade de acesso a variáveis comuns a ambos, sendo que suas áreas de memórias são diferentes e isoladas. Como desvantagem o uso da comunicação pelo meio núcleo pode causar ineficiência para casos de frequência constante e volume dados a serem transmitidos serem muito grandes. Para isso surgiu a memória compartilhada, servindo casos que necessitem ter uma área de memória conhecida por processos que têm interesse em acessá-la sem precisar do núcleo como intermediário. A imagem ao lado representa um exemplo de memória compartilhada entre 2 processos P_a e P_b em um sistema operacional UNIX:

O processo P_a realiza solicitação ao núcleo para criação de área de memória compartilhada, o núcleo responde alocando uma nova área de memória e registrando na sua lista de memórias compartilhadas e retorna uma mensagem que contém o identificador da área alocada para o processo P_a . O processo P_a requisita ao núcleo que área identificada pelo id seja adicionada ao seu espaço de endereçamento, o processo de modificar a configuração de memória do processo P_a para adicionar a área indicada pelo id em seu espaço de endereçamento é feita pelo núcleo e retorna para o processo P_a um ponteiro para a área de memória alocada, o processo P_b realiza os mesmos procedimentos de solicitação e tem sua configuração de memória modificada pelo núcleo e recebimento de um ponteiro para nova área de memória alocada e no final os processos P_a e P_b já podem se comunicar através de escritas e leituras dentro da área de memória compartilhada.



2. Classifique as filas de mensagens UNIX de acordo com os tipos de comunicação discutidos no Capítulo 8.

As filas são uma comunicação confiável entre processos de N:M ou N:1 indireta, orientado a mensagem, capacidade finita (depende do tipo do sistema operacional possui um valor) e para diferentes tipos de projetos a fim de realizar a comunicação de forma assíncrona ou síncrona.

3. Classifique os pipes UNIX de acordo com os tipos de comunicação discutidos no Capítulo 8.

Os pipes são canais de comunicação entre dois processos de forma indireta, comunicação síncrona, unidirecional, orientada por fluxo e possui capacidade finita.

4. Classifique as áreas de memória compartilhadas de acordo com os tipos de comunicação discutidos no Capítulo 8.

A memória compartilhada é uma área onde vários processos podem acessar de maneira simultânea sendo N:M ou N:1, ocorrendo comunicação de forma indireta, pois não existe necessidade de identificar o processo que está lendo ou escrevendo.

5. Sobre as afirmações a seguir, relativas aos mecanismos de comunicação, indique quais são incorretas, justificando sua resposta:

(a) As filas de mensagens POSIX são um exemplo de canal de comunicação com capacidade nula.

FALSO: as filas possuem capacidade finita.

(b) A memória compartilhada provê mecanismos de sincronização para facilitar a comunicação entre os processos.

FALSO: O sistema operacional que provê os mecanismos para poder realizar a sincronização entre processos pois a comunicação com a memória compartilhada é feita assíncrona entre os processos.

(c) A troca de dados através de memória compartilhada é mais adequada para a comunicação em rede.

FALSO: As suas áreas de memória são distintas para comunicação em rede.

(d) Processos que se comunicam por memória compartilhada podem acessar a mesma área da RAM.

(e) Os pipes Unix são um bom exemplo de comunicação M:N.

FALSO: os pipes realizam comunicação do tipo 1:1.

(f) A comunicação através de memória compartilhada é particularmente indicada para compartilhar grandes volumes de dados entre dois ou mais processos.

(g) As filas de mensagens POSIX são um bom exemplo de canal de eventos.

(h) Nas filas de mensagens POSIX, as mensagens transitam através de arquivos em disco criados especialmente para essa finalidade.

FALSO: Somente a memória do kernel é utilizada para troca de mensagens.

(i) Em UNIX, um pipe é um canal de comunicação unidirecional que liga a saída padrão de um processo à entrada padrão de outro.