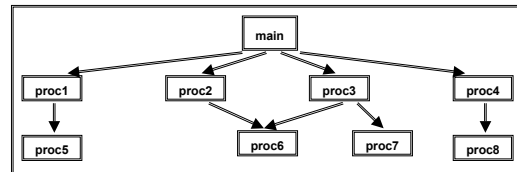


## Chamadas Remotas de Procedimentos (RPC)

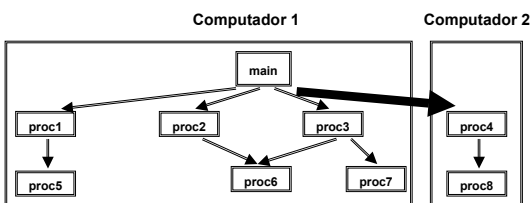
- Chamada Remota de Procedimento (RPC) ou Chamada de Função ou Chamada de Subrotina
- Método de transferência de controle de parte de um processo para outra parte
- Procedimentos => permite a divisão do programas em vários pedaços

## O Conceito de Procedimentos



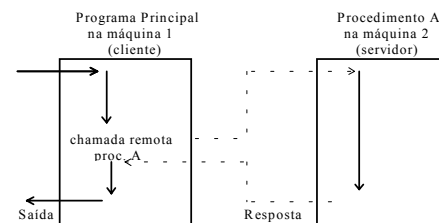
- Um programa convencional consiste de um ou mais procedimentos, geralmente organizados em uma hierarquia de chamadas.
- Uma seta de um procedimento  $n$  para um procedimento  $m$  significa uma chamada de  $n$  para  $m$

## RPC: Programa Distribuído



- A divisão ocorre entre o programa principal e o procedimento 4.
- Um protocolo de comunicação é necessário para implementar a chamada remota.

## RPC: Modelo de Execução

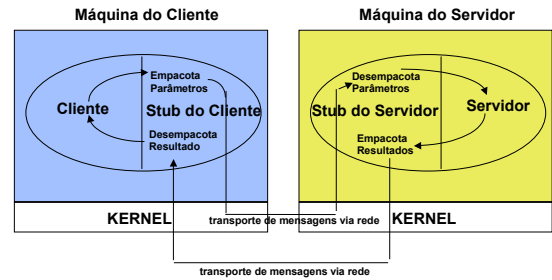


- O processo cliente fica bloqueado durante a execução do procedimento remoto!

## RPC

- **Objetivo:** Tornar mais fácil a implementação de Aplicações Distribuídas
- Esconde o código de chamadas a rede em procedimentos chamados *stubs*
  - *Stubs* -> procedimentos que contêm o código de chamadas a rede.
  - Com *stubs* o RPC protege os programas de aplicação (cliente e servidor) de preocupações com detalhes como sockets.
  - O RPC inclui uma especificação para formato padrão dos dados (visando interoperabilidade), e nos *stubs* acontece a conversão dos dados
    - No RPC da Sun o padrão para a representação dos dados é o XDR (*eXternal Data Representation Standard*)
  - Os *stubs* são gerados automaticamente por algum compilador. Exemplo: O RPCGen da Sun

## Passos de uma Chamada Remota de Procedimentos



## Chamadas Remotas de Procedimentos (RPC)

- Idéia do modelo é estender o conceito de chamada de procedimento convencional para ambientes distribuídos.
  - a ênfase é em distribuição e não em concorrência!
  - objetivo é simplificar a programação distribuída, tornando-a semelhante à programação convencional!
- *Remote Procedure Call* (RPC): subrotina chamada pode ser função ou procedimento
- Procedimentos => permitem a divisão dos programas em vários pedaços que podem executar em máquinas arbitrárias.

## Semântica de Chamadas Remotas

- Ao contrário do que acontece em chamadas convencionais, uma das máquinas envolvidas pode falhar e a outra permanecer ativa...
- Sistema pode garantir que a execução da rotina chamada ocorre:
  - exatamente uma vez
    - difícil!!
  - no mínimo uma vez
    - operações devem ser idempotentes!
  - no máximo uma vez
- Sistema deve retornar indicações no caso de máquina servidora falhar!

## RPC - Implementação

- O código das chamadas a rede é escondido em procedimentos chamados *stubs*
  - *Stubs* -> procedimentos que contêm o código de chamadas a rede.
  - Com *stubs* o RPC protege os programas de aplicação (cliente e servidor) de preocupações com detalhes como sockets.
- Cabe aos stubs a passagem de parâmetros entre procedimentos.
  - Máquinas diferentes podem usar representações diferentes de dados como inteiros, caracteres, etc.
  - O que fazer com dados complexos, como listas, etc?

## Um exemplo de RPC: Sun-RPC

- Sistema originalmente criado para máquinas Sun.
  - oferecido atualmente em diversos sistemas operacionais!
- A arquitetura definida inclui:
  - uma linguagem para definição das interfaces (cabeçalhos de procedimentos, etc);
  - a ferramenta RPCGEN, que gera os stubs cliente e servidor automaticamente;
  - uma biblioteca RPC, que pode ser usada diretamente na construção de programas que não usem o RPCGEN;
  - o protocolo de comunicação entre os stubs.
- Pode utilizar TCP ou UDP

## Sun-RPC - Tradução de dados

- Tradução entre formatos de dados: utilização de uma representação padrão, XDR (*eXternal Data Representation Standard*).



- A conversão é especificada para um conjunto pré-definido de tipos de dados.