KOMUNIKÁCIA MEDZI PROCESMI

Spolupracujúce procesy

- Procesy môžu byť
  - nezávislé procesy alebo
  - spolupracujúce procesy.
- Príčiny spolupráce
  - Zdieľanie informácií
  - Zrýchlenie výpočtov (program je rýchlejší ak sa vykonáva paralelne v niekoľkých podúlohach)
  - Modularita
  - Výhoda (editovanie, tlač, kompilácia)

Spolupracujúce procesy pomocou správ

- IPC (Interprocess communication)
- IPC poskytuje mechanizmy pre komunikáciu a synchronizáciu činnosti.
- Posielanie správ a zdieľaná pamäť nie sú vzájomne sa vylučujúce, môžu sa používať súčasne v rámci jedného procesu a systému
- Základná štruktúra systému zasielania správ:
  - dve základné operácie : **send(message)**, **receive(message)**;
- Procesy P a Q komunikujú cez komunikačné linky.

Správy

- Linky medzi procesmi sú jednosmerné – proces môže buď posielať alebo dostavať správu
- Správa môže mať buď *pevnú* alebo v*ariabilnú* dĺžku.
  - **Pevná dĺžka** správy dovoľuje jednoduchú fyzikálnu implementáciu, ale program musí deliť správu na rovnako veľké časti
  - **Variabilná dĺ**žka správy umožňuje jednoduchšie programovanie, ale fyzikálna implementácia je zložitejšia

Implementácie prepojenia a operácií *send* a *receive*

- ***priama*** alebo ***nepriama*** komunikácia,
- ***symetrická*** alebo ***asymetrická*** komunikácia,
- ***automatické*** alebo ***explicitné*** bufrovanie,
- vyslanie ***kópiou*** alebo ***odkazom***,
- ***pevná*** alebo ***variabilná*** dĺžka správy.

Synchronizácia komunikácie

- Zasielanie správ môže byť blokujúce alebo neblokujúce.
- **Blokujúce** je synchrónne zasielanie správ
- **Neblokujúce** je asynchrónne zasielanie správ
- Operácie **send** a **receive** môžu byť buď blokujúce alebo neblokujúce

Priama komunikácia - pomenovanie

Priama komunikácia

- ***send*** *(**P**, správa)* - zasiela správu procesu P,
- ***receive*** *(**Q**, správa)* - prijíma správu od procesu Q.

- Vlastnosti spojenia
  - nadväzuje automaticky medzi každou dvojicou procesov, ktoré chcú komunikovať,
  - spojenie sa nadväzuje medzi dvomi procesmi,
  - medzi každým párom komunikujúcich procesov existuje jedno spojenie,
  - linka môže byť jednosmerná, ale väčšinou je obojsmerná.

Príklad: **Úloha Producent-Konzument pomocou správ**

Proces-producent je definovaný nasledovne:

repeat

pripraví položku v *nextp*

send*(konzument, nextp)*

until false;

Proces-konzument je definovaný nasledovne:

repeat

receive*(producent,nextc);*

spracuje položku

until *false;*

Typy priamej komunikácie

- ***Symetrická*** - vysielajúci aj prijímajúci proces uvádzajú meno svojho partnera.
- ***Asymetrická*** - len vysielajúci proces uvádza meno adresáta.
- Operácie *send* a *receive* majú v tomto prípade tvar:
  - ***send*** *(P, správa)* - zasiela správu procesu P,
  - ***receive*** *(id, správa)* - prijíma správu od ľubovoľného procesu;

  id je premenná, ktorá obsahuje identifikátor procesu, s ktorým bolo nadviazané spojenie.

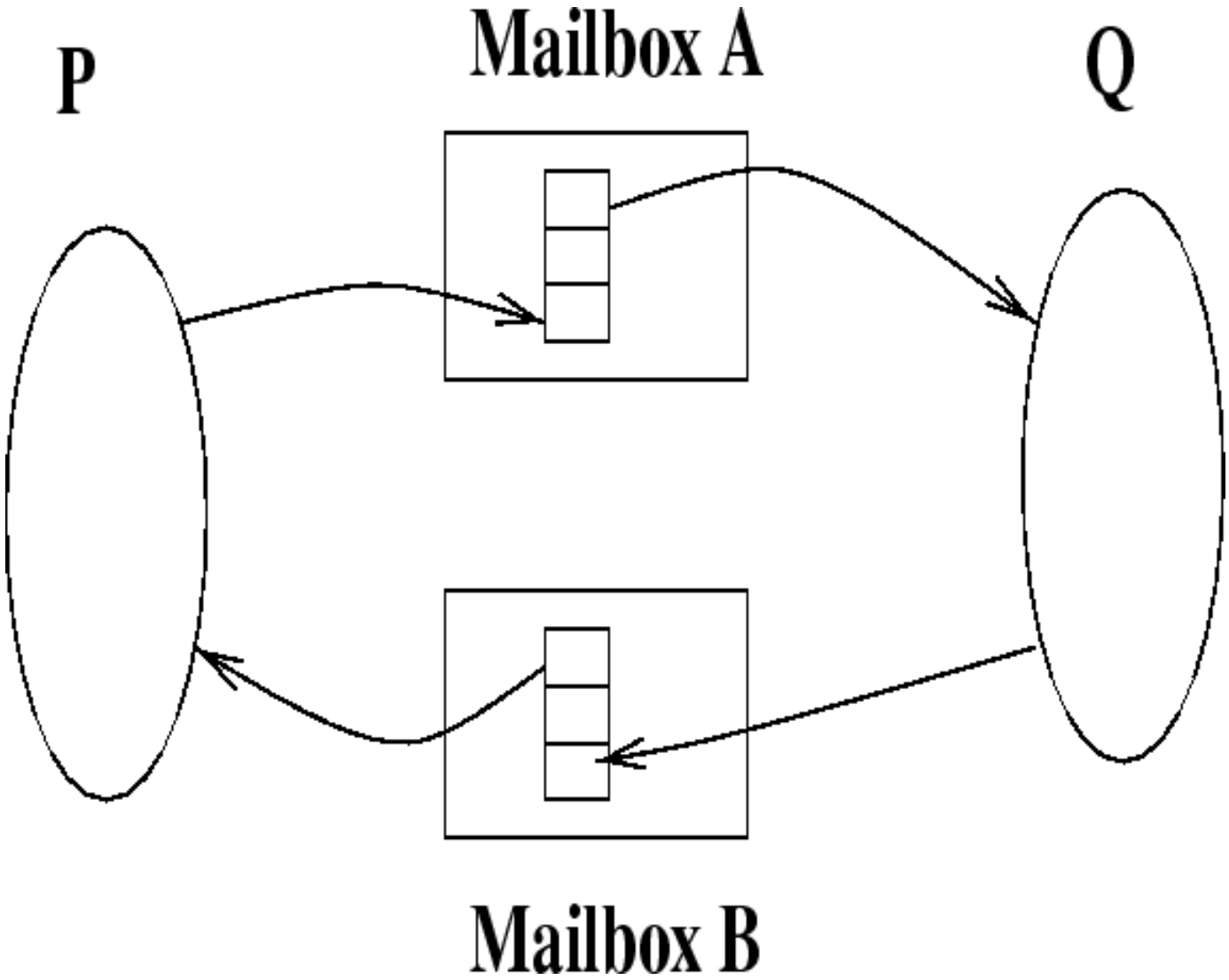
Nepriama komunikácia - pomenovanie

Vysielanie a príjem správ z ***mailbox-***ov (schránok). Sú to abstraktné objekty, do ktorých sa môže vložiť správa alebo vybrať

send *(**A**, správa)* - zasiela správu do mailbox-u ***A***,
receive *(**A**, správa)* - prijíma správu od mailbox-u ***A***.

Komunikačné spojenie v tomto prípade má nasledovné vlastnosti:

- spojenie sa nadväzuje,
  - len ak procesy majú prístup k zdieľanému mailbox-u,
  - medzi viacerými procesmi,
  - medzi každým párom komunikujúcich procesov je viac komunikačných liniek a každá zodpovedá jednému mailbox-u,
  - linka môže byť jednosmerná alebo obojsmerná.


P
Mailbox A
Q
Mailbox B

Mailboxy pokr.

- Vlastníctvo mailbox-ov
  - **proces**
    - deklaruje premennú typu mailbox
    - keď sa ukončí proces, zruší sa aj mailbox
    - každý proces, ktorý vie meno mailboxu ho môže použiť
  - **systém**
    - vytvorí mailbox
    - posiela a získava správy z mailbox-u
    - zruší mailbox

Bufrovanie

- Komunikačná linka má **kapacitu** - určuje, aký počet správ sa v nej môže nachádzať, k linke je pripojený front správ.
- Tri spôsoby implementácie frontov:
  - **S nulovou kapacitou.** Max. dĺžka frontu je 0, t.j. vo fronte nemôže čakať žiadna správa, vysielajúci proces musí počkať, až prijímajúci proces preberie správu. Procesy sa musia zosynchronizovať pri odovzdávaní správy - ***rendezvous.***
  - **S obmedzenou kapacitou**. Front má konečnú dĺžku ***n***, t.j. môže v ňom čakať najviac ***n*** správ. Ak front nie je plný, prichádzajúca správa sa doň zaradí a vysielajúci proces môže pokračovať vo svojom vykonaní bez čakania. Ak front je plný, vysielajúci proces je zablokovaný, pokiaľ sa uvoľní miesto.

Bufrovanie pokr.

- Pri automatickom bufrovaní odosielateľ nevie čí správa bola doručená – potrebné potvrdzovanie (ACKNOWLEDGMENT)

Proces ***P*** vykoná:

send *(Q, správa)* - zasiela správu Q

receive *(Q, potvrdenie)* - prijíma potvrdenie od Q

Proces **Q** vykoná:

receive *(P, správa)* - prijíma správu od *P*

send *(P, potvrdenie)* - zasiela potvrdenie *P*

Bufrovanie pokr.

- **Iné alternatívy**
  - **Odosielateľ sa neblokuje nikdy :** v prípade bez bufrovania, to znamená stratu správy ak procesy nie sú synchronizované.

    send(Q, *sprava1*) →***receive***(P, *sprava1*)

    send(Q, *sprava2*) → ???

    send)Q, *sprava3*) → ***receive***(P, *sprava3*)
  - **Zaslanie správy len ak príde potvrdenie :**

    send(Q, *sprava1*) → ***receive***(P, *sprava*)

    (Proces P čaká)

    receive(Q, *sprava*) ← ***reply***(P, *sprava*)

    (Proces Q sa nezablokuje)

Problémy pri zasielaní správ

- Potrebná obsluha chybových stavov - *Exception-condition handling.*
  - Ukončenie procesu
    - *P* čaká na správu od Q, ktorý bol ukončený : *P* bude zablokovaný donekonečna. OS musí buď upozorniť *P*, že Q bol ukončený, alebo ho ukončiť.
    - *P* zasiela správu Q, ktorý bol ukončený.
      - pri automatickom bufrovaní – bez problémov pre P. Ak *P* potrebuje potvrdenie o prijatí, musí si to **explicitne** naprogramovať.
      - bez bufrovania, *P* bude zablokovaný natrvalo. OS musí buď upozorniť alebo ukončiť proces

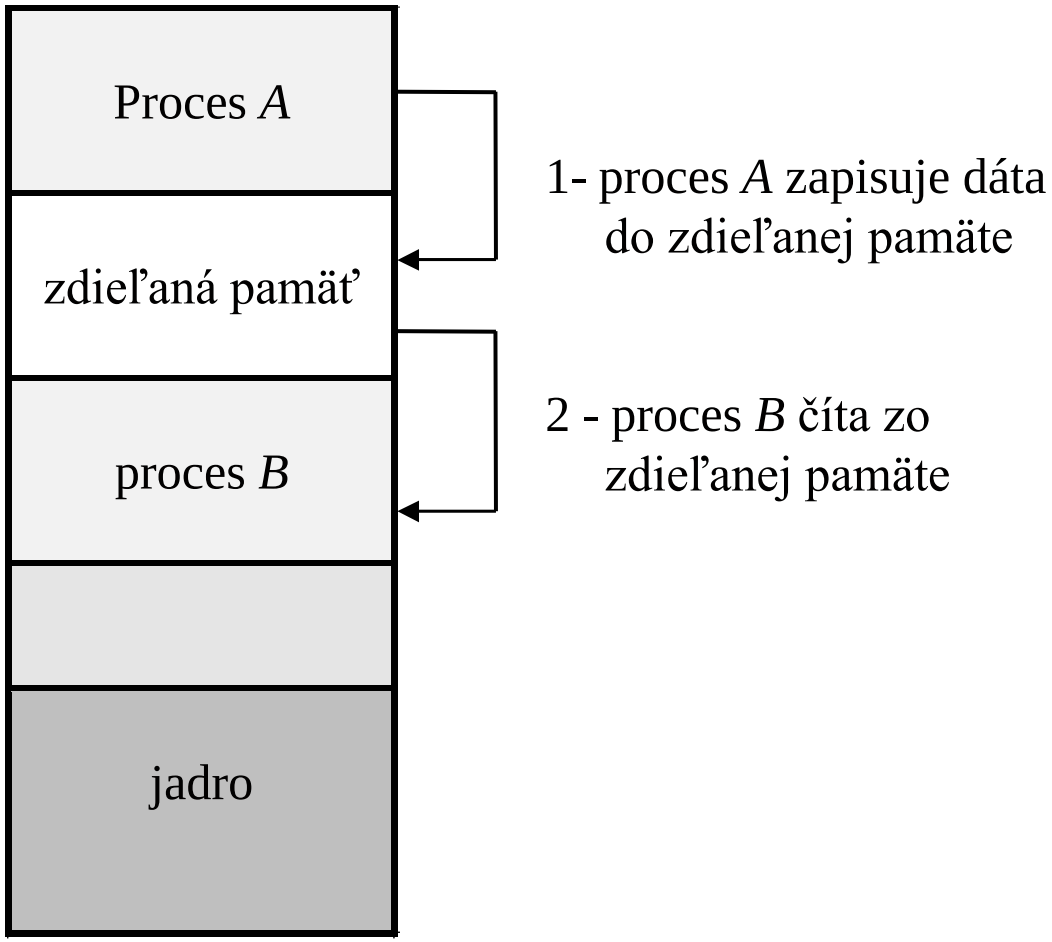
Problémy pri zasielaní správ pokr.

- **Stratená správa**
  - Používajú sa nasledovné metódy ošetrenia tejto udalosti:
    1. ***OS je zodpovedný*** za zistenie straty správy a jej opätovného vyslania.
    2. Za zistenie straty správy a jej opätovného vyslania je ***zodpovedný vysielajúci proces.***
    3. ***OS je zodpovedný*** za zistenie straty správy a upozornenie vysielajúceho procesu. ***Opätovné vyslanie správy je prenechané vysielajúcemu procesu.***
- **Poškodená správa**
  - OS obyčajne zistí poškodenie pomocou rôznych kontrol (kontrolné sumy, parita, **C**yclic **R**edundancy **C**ode a iné) a zabezpečí aj opätovné vyslanie správy.

Zdieľaná pamäť

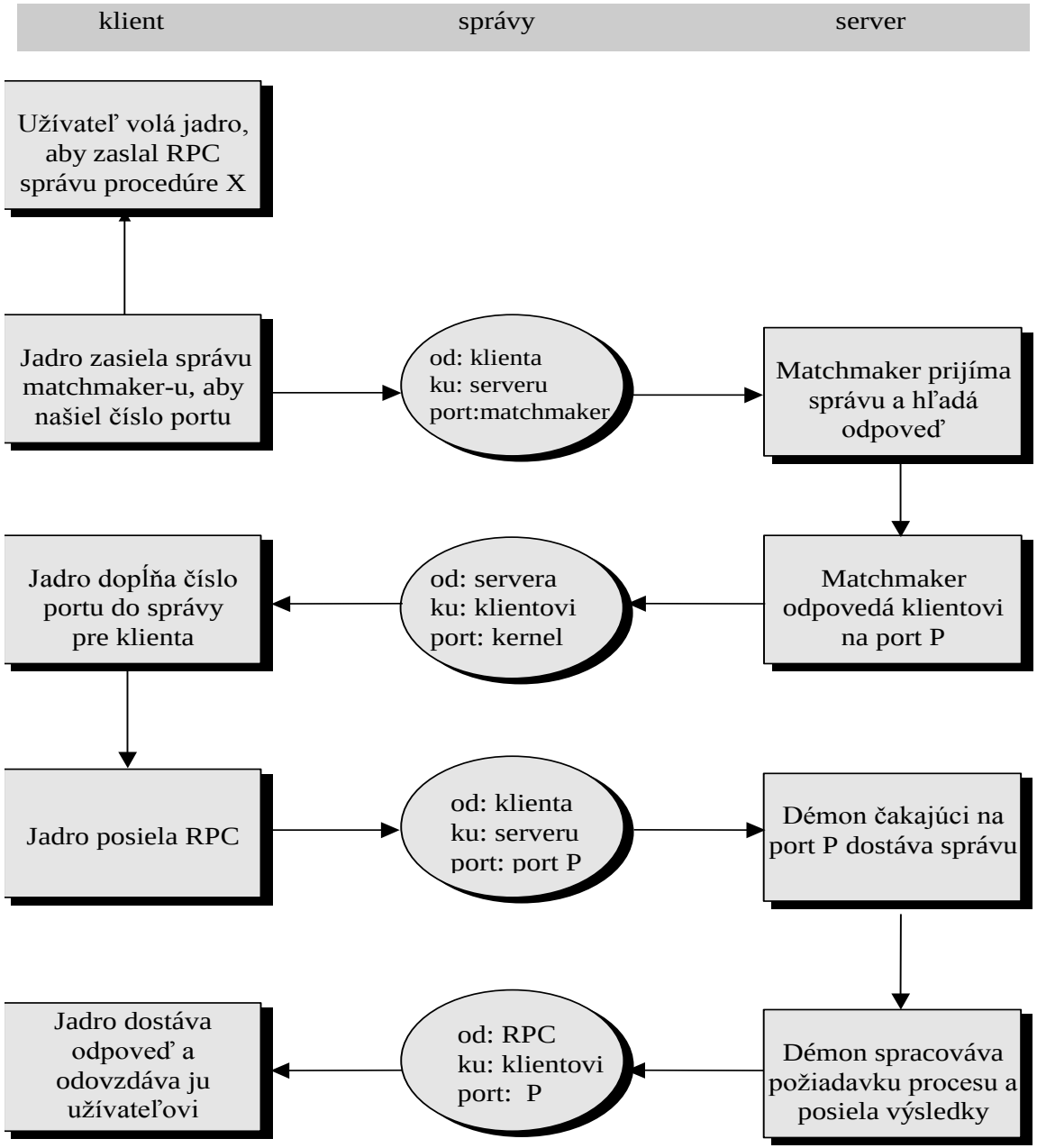
- **Najrýchlejšia komunikácia medzi procesmi**
- Ten istý pamäťový segment je mapovaný do adresných priestorov dvoch alebo viacerých procesov
- Potrebná synchronizácie prístupu pri súbežnom prístupe k zdieľaným dátam.

Zdieľaná pamäť pokr.



Iné alternatívy pokr.

- **Volanie vzdialenej procedúry (RPC) –** pre distribuované systémy
  - Rozšírenie synchrónnej komunikácie procedúry (Remote Procedure Call - RPC).
  - mechanizmus, podobný mechanizmu volania lokálnej procedúry
  - procesy takejto aplikácie sa vykonávajú na rôznych uzloch siete a komunikujú pomocou zasielania správ.
  - pre zabezpečenie komunikácie medzi procesmi - potrebné správu adresovať na konkrétny ***port*** (id. číslo koncového bodu spojenia), ktorý je pridelený príslušnej procedúre.
  - Zistenie portu sa uskutočňuje
    - *staticky* – adresa je pridelená ešte pri kompilácií
    - *dynamicky* - adresa sa zisťuje pred nadviazaním spojenia.



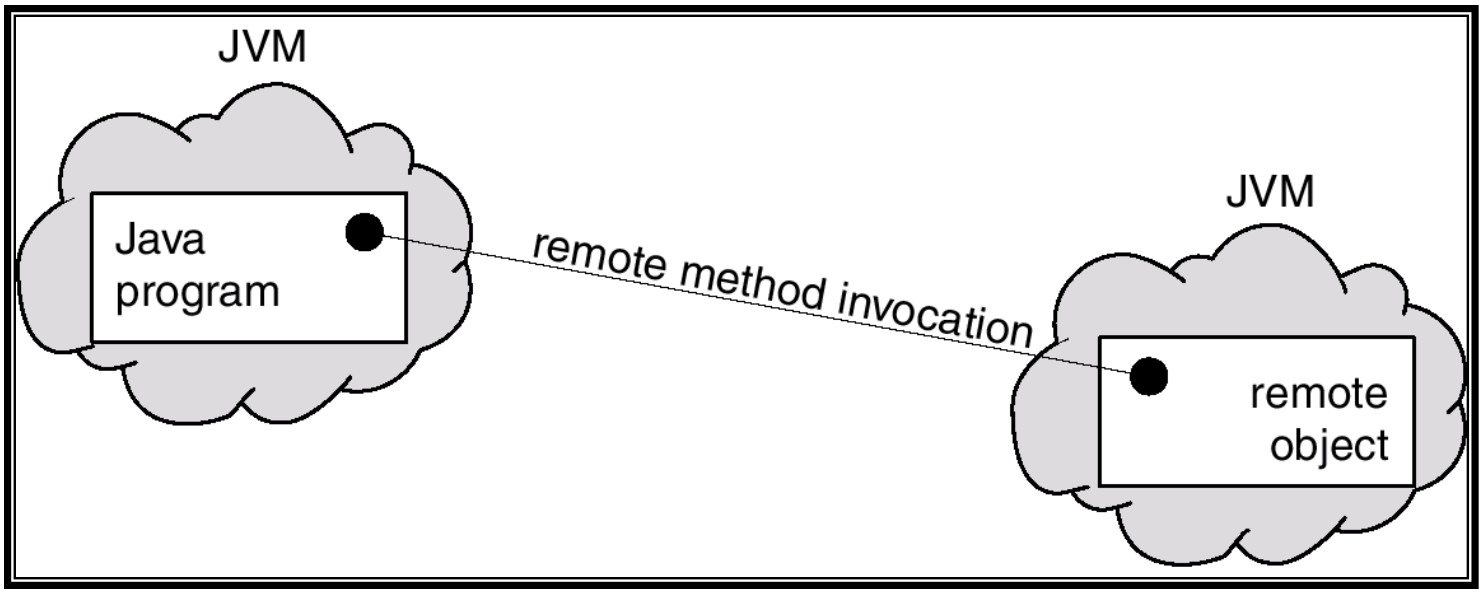
Vykonanie volania vzdialenej procedúry (RPC)

RPC pokr.

- Ak sa procesy, využívajúce RPC vykonávajú na heterogénnych strojoch - problém, ako si majú vymieňať dáta (rozdielne reprezentácie dát).
- protokol ***XDR (eXternal Data Representation protocol),***
  - zabezpečuje zakódovanie jednotlivých dát

Iné prostriedky komunikácie

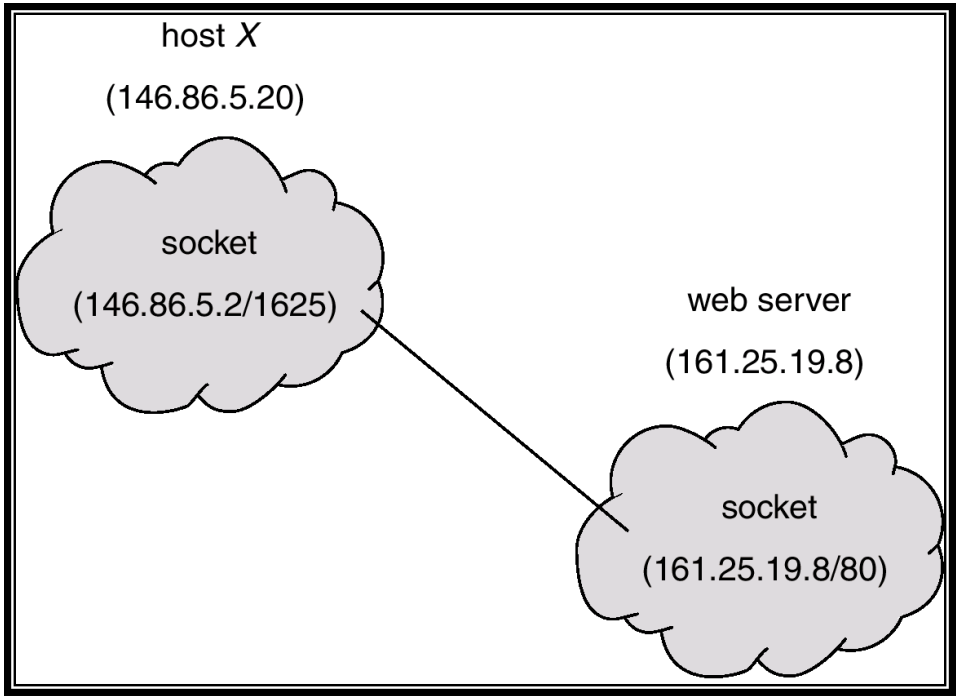
- RMI (Remote Method Invocation) - Java



- Rúry – nepomenované (medzi príbuznými procesmi), pomenované
- Signály – informácia o výskyte určitej udalosti

Iné prostriedky komunikácie

- **Schránky** (sockets)– komunikácia v počítačovej sieti



- **MPI – knižnica** pre komunikáciu medzi procesmi distribuovaných systémov