

# TCP/IPとプロセス間通信

2007年1月12日

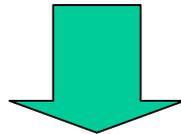
海谷 治彦

# 目次

- 動機
- 通信路の識別 IPアドレスとポート番号
- ソケット通信の基本
  - 一対一通信の例
- サーバーを中心としたスター型の通信

# 動機

- 今までの講義でpipe等を使い, 親子関係のあるプロセス間で通信(read/write)を行うことができた.
- しかし, 赤の他人とは通信がこれではできない.
- もっと一般化して他のOS(マシン)上のプロセスとも通信ができない.



できるような技術を学びましょう！

# ソケットによるプロセス間通信

- 同じマシンは勿論, 異なるマシン間のプロセスの通信を可能にする仕組み.
  - 無論, 意図的に通信を禁止してなければ.
- マシンは通常IPアドレスで識別されている.
  - `www.cs.shinshu-u.ac.jp` 等も内部的にはIPアドレスで管理されている.
- 通信路はポート番号で識別されている.
  - 若い番号は特定用途に予約されている.
    - 25 メール 80 ウェブ等
- IPアドレス+ポート番号の対で通信路を特定して, プロセス間通信を行う.

# 基本知識・TIPS

- 自分のマシンのIPアドレスの調べ方  
/sbin/ifconfig コマンドを利用.
- ちなみに以下のアドレスは自分自身を指す.  
127.0.0.1
- 同一番号のポートはある程度, 時間を置かないと再度の利用できません.

# 以下のサンプルの基本仕様 1

- server?.c サーバー
  - 相手からの接続を待つプロセスを生成するプログラム.
  - 接続を待つポート番号を指定する必要がある.
- client?.c クライアント
  - 接続を待つサーバー型のプロセスに接続を行うプロセスを生成するプログラム.
  - サーバーのいるマシンのアドレスとポート番号を指定する必要がある.

# 基本仕様 2

- 通信はTCP/IPで行う.
  - 無論, それ以外のプロトコルも存在するが. 例えば, PF\_APPLETALK 等.
  - PF\_INET SOCK\_STREAM の組合せが TCP/IPに相当.
- ホストアドレス指定もIP型で行う.
  - 無論, それ以外の指定方法もある, 例えば, AF\_APPLETALK 等.
  - AF\_INET がIPアドレス型の指定に相当.

# 例1 server1/client1

- 単にサーバーが特定の文字列をクライアントに送信するプログラム.
  - サーバーが書き
  - クライアントが読み
- 多分, もっとも簡単.
- サーバーは通信準備に4ステップ
- クライアントは準備に2ステップ
  
- 詳細はサンプルコードを参照.
- 異なるマシン間でも通信は成立する(はず).

## 例2 server2/client2

- 読み書きを逆にする
  - サーバーが読み
  - クライアントが書き
- 要はTCP/IPのソケットは読み書き双方向通信.
- 全二重 (full-duplex)と呼ばれる性質.

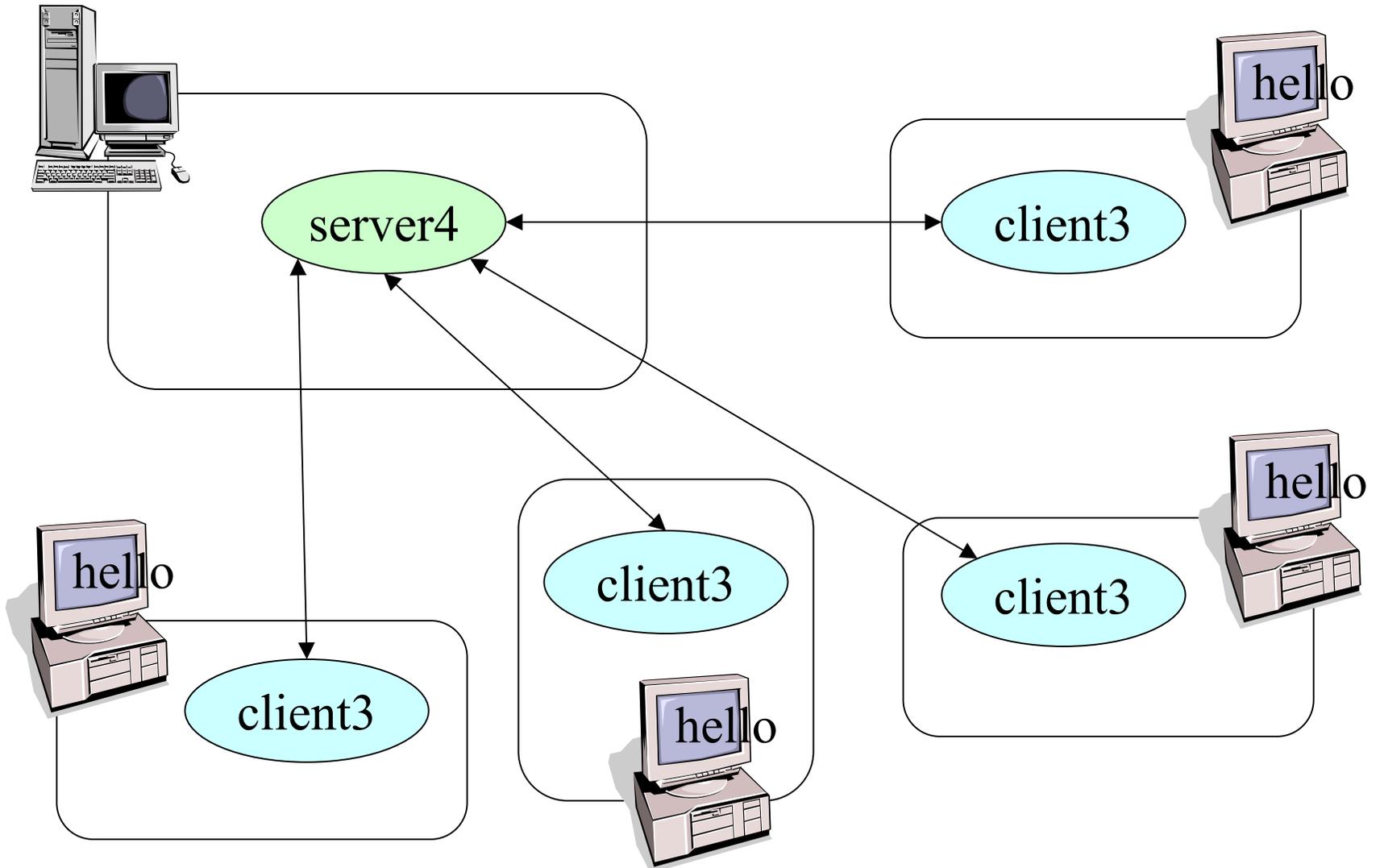
# 例3 簡単なテキストチャット

- 双方向でテキストチャットができるはず.
- コントロールD等でどちらかが終了すればセッションは終了する.
- pthread を使ってるんで, 割と楽勝.
  - スレッド無しで作るのは相当シンドい.
- 共有関数があるので以下とか  
gcc common3.c server3.c -o s  
gcc common3.c client3.c -o c

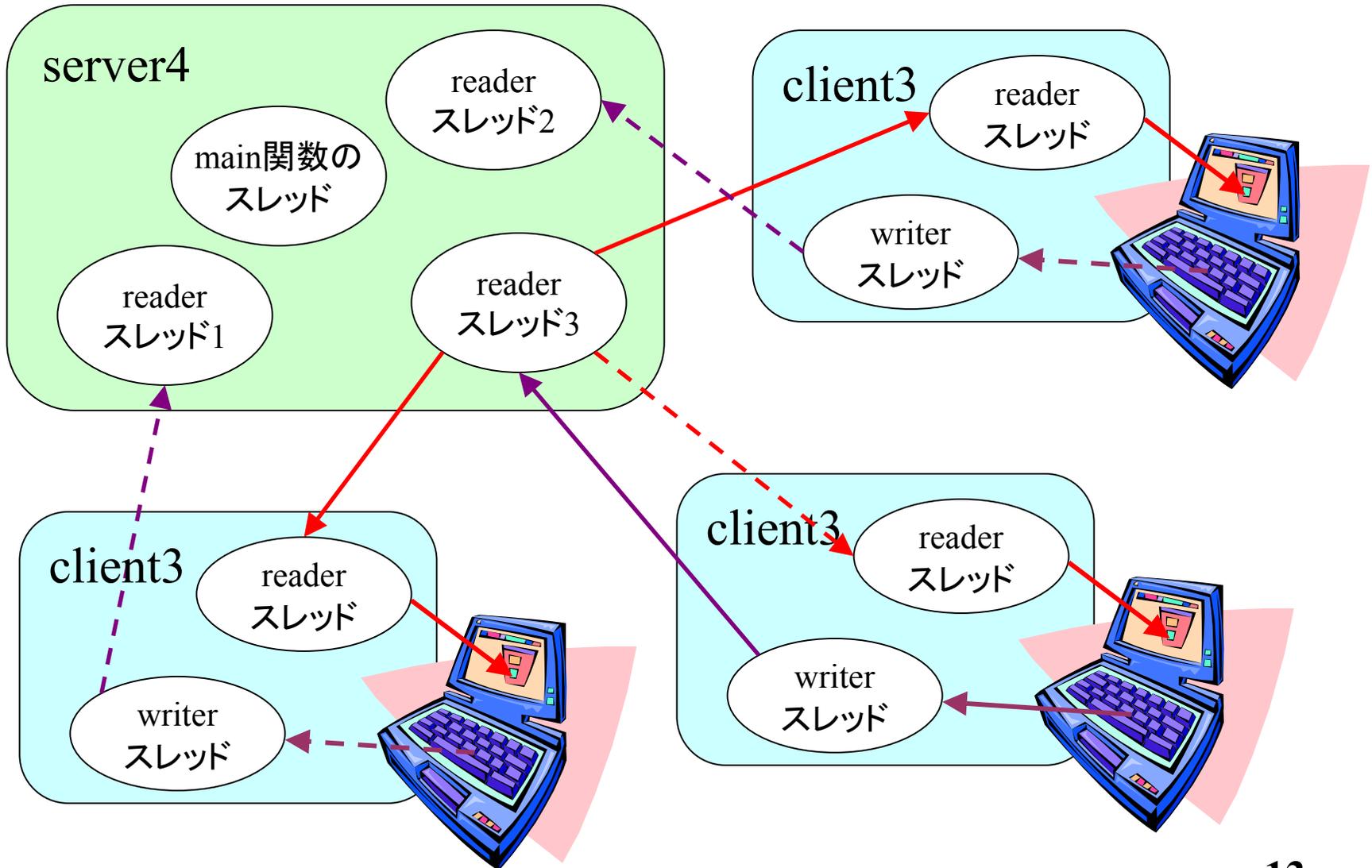
## 例4 多人数のテキストチャット

- サーバー(server4.c)を中心として、クライアント(client3.c)で複数接続するスター型構造のプロセス間通信構造.
- サーバー側ではクライアントが接続される毎にスレッドを走らせて対応を行う構造.
  - コレもスレッド無しで開発すると大変.

# 例4の構造イラスト



# スレッドの構造



# 例5 クライアントの識別

- server5.c 内容は server4.c とほぼ同じ.
- サーバーからクライアントの素性がわかれば, 簡単に接続制限等を行うことができる.

# 便利な関数群

- htons, htonl, htobs, ntohl
  - hはホストの意味
  - nはネットワーク・バイトオーダー(big endian)
  - sは16bit (short), lは32bit(long)
  - バイトオーダーの違いを吸収するための関数群.
- inet\_aton
  - 160.252.120.10等の表記をバイナリ表現に変換.
- inet\_ntoa
  - 上記の逆.
- 詳細はオンラインマニュアルとサンプルを参照.