

Sockets

Bruno Pereira

Universidade Federal de Minas Gerais

bruno.ps@live.com

18 de setembro de 2016

Roteiro

- 1 Introdução
 - Entendendo Sockets

- 2 Programação
 - Cliente e Servidor

- 3 Extra

Introdução

Histórico

- Década de 1980 ARPA^a, deu à Berkeley a tarefa de criar um SO que desse suporte a ARPAnet
- Unix BSD (Berkeley Software Distribution)
- A Berkeley Sockets Interface é a base para a maioria das interfaces entre protocolos de internet TCP/IP existentes.

^aAdvanced Research Projects Agency of the Department of Defense

Introdução

Histórico

- Década de 1980 ARPA^a, deu à Berkeley a tarefa de criar um SO que desse suporte a ARPAnet
- Unix BSD (Berkeley Software Distribution)
 - A Berkeley Sockets Interface é a base para a maioria das interfaces entre protocolos de internet TCP/IP existentes.

^aAdvanced Research Projects Agency of the Department of Defense

Introdução

Histórico

- Década de 1980 ARPA^a, deu à Berkeley a tarefa de criar um SO que desse suporte a ARPAnet
- Unix BSD (Berkeley Software Distribution)
- A Berkeley Sockets Interface é a base para a maioria das interfaces entre protocolos de internet TCP/IP existentes.

^aAdvanced Research Projects Agency of the Department of Defense

Sockets

- Sockets é uma maneira de falar com outros programas remotos usando arquivos descritores do Unix
- O que são arquivos descritores?
 - "...Ok—you may have heard some Unix hacker state, 'Jeez, everything in Unix is a file!' ..." ^a
 - É um inteiro associado a um arquivo FIFO, pipe, terminal, leitura do disco, **conexão**

^aBeej's Guide to Network Programming

Sockets

- Sockets é uma maneira de falar com outros programas remotos usando **arquivos descritores do Unix**
- O que são arquivos descritores?
 - "...Ok—you may have heard some Unix hacker state, 'Jeez, everything in Unix is a file!' ..." ^a
 - É um inteiro associado a um arquivo FIFO, pipe, terminal, leitura do disco, **conexão**

^aBeej's Guide to Network Programming

Sockets

- Como fazer para criar um descritor para uma conexão?
- Os OS possuem algumas rotinas para esta finalidade
- `socket(...)` - vai retornar um descritor para uma conexão ^a
 - `send(...)`
 - `recv(...)`

^acomando: `man socket, send, recv`

Sockets

- Como fazer para criar um descritor para uma conexão?
- Os OS possuem algumas rotinas para esta finalidade
- `socket(...)` - vai retornar um descritor para uma conexão ^a
 - `send(...)`
 - `recv(...)`

^acomando: `man socket, send, recv`

Sockets

- Como fazer para criar um descritor para uma conexão?
- Os OS possuem algumas rotinas para esta finalidade
- **socket(...)** - vai retornar um descritor para uma conexão ^a
 - send(...)
 - recv(...)

^acomando: man socket, send, recv

Sockets - Nomenclatura

- Existe uma nomenclatura para cada um dos "lados" da comunicação
 - Servidor
 - Espera por conexões de entrada
 - Fornece certos tipos de serviços para a outra parte
 - Cliente
 - Solicita uma conexão ao servidor
 - Faz requisições de serviços
- Não é o computador quem decide quem é cliente ou quem é servidor, mas sim a forma como o programa usa os sockets.

Sockets - Nomenclatura

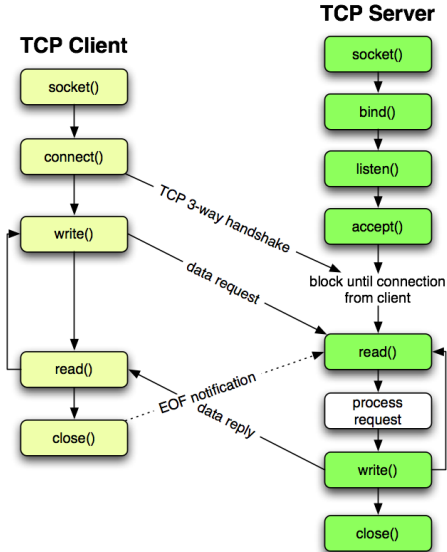
- Existe uma nomenclatura para cada um dos "lados" da comunicação
- Servidor
 - Espera por conexões de entrada
 - Fornece certos tipos de serviços para a outra parte
- Cliente
 - Solicita uma conexão ao servidor
 - Faz requisições de serviços
- Não é o computador quem decide quem é cliente ou quem é servidor, mas sim a forma como o programa usa os sockets.

Sockets - Nomenclatura

- Existe uma nomenclatura para cada um dos "lados" da comunicação
- Servidor
 - Espera por conexões de entrada
 - Fornece certos tipos de serviços para a outra parte
- Cliente
 - Solicita uma conexão ao servidor
 - Faz requisições de serviços
- Não é o computador quem decide quem é cliente ou quem é servidor, mas sim a forma como o programa usa os sockets.

Sockets - Nomenclatura

- Existe uma nomenclatura para cada um dos "lados" da comunicação
- Servidor
 - Espera por conexões de entrada
 - Fornece certos tipos de serviços para a outra parte
- Cliente
 - Solicita uma conexão ao servidor
 - Faz requisições de serviços
- Não é o computador quem decide quem é cliente ou quem é servidor, mas sim a forma como o programa usa os sockets.



Sockets - Tipos

- Stream Sockets - SOCK_STREAM
 - Usa o protocolo TCP/IP ^a
 - Mantém uma conexão confiável
 - Os dados vão em ordem
 - Livre de erros

^aThe Transmission Control Protocol RFC 2001[6]. Internet Protocol address.

Sockets - Tipos

- Datagram Sockets - SOCK_DGRAM
 - Usa o protocolo UDP/IP ^a
 - **Não** mantém uma conexão confiável
 - **Não** garante a chegada em ordem dos dados

^aUser Datagram Protocol RFC 768[5].

Sockets - Tipos

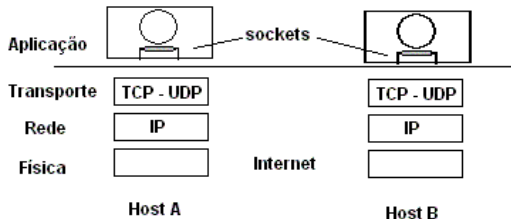
- Datagram Sockets - SOCK_DGRAM
 - Usa o protocolo UDP/IP ^a
 - **Não** mantém uma conexão confiável
 - **Não** garante a chegada em ordem dos dados

^aUser Datagram Protocol RFC 768[5].

Ethernet IP UDP TFTP **Data**

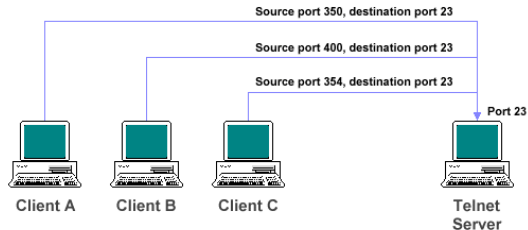
Sockets - Porta

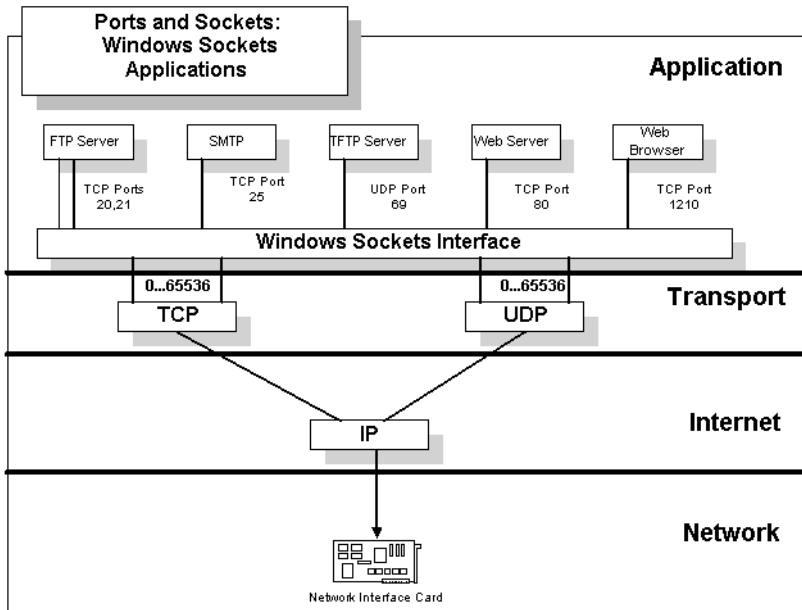
- Como saber qual aplicação devo enviar os dados?
- Cada aplicação vai ter uma porta associada
 - Comando: `less /etc/services`
 - `netstat -antp`



Sockets - Porta

- Como saber qual aplicação devo enviar os dados?
- Cada aplicação vai ter uma porta associada
 - Comando: `less /etc/services`
 - `netstat -antp`





Cliente e Servidor

Example

Servidor.c

- Veja o código Servidor.c [3]
- Comando: nc <endereço> <porta>
 - Comando: nc localhost 5050

Example

Cliente.c

Cliente e Servidor

Example

Servidor.c

- Veja o código Servidor.c [3]
- Comando: nc <endereço> <porta>
 - Comando: nc localhost 5050

Example

Cliente.c

Cliente e Servidor

Example

Servidor.c

- Veja o código Servidor.c [3]
- Comando: nc <endereço> <porta>
 - Comando: nc localhost 5050

Example

Cliente.c

- Veja o código Cliente.c [3]
- Execute uma instância do servidor e uma do cliente

Cliente e Servidor

Example

Servidor.c

- Veja o código Servidor.c [3]
- Comando: nc <endereço> <porta>
 - Comando: nc localhost 5050

Example

Cliente.c

- Veja o código Cliente.c [3]
- Execute uma instância do servidor e uma do cliente



Brian "Beej Jorgensen" Hall.

Beej's Guide to Network Programming.

<http://beej.us/guide/bgnet/output/html/multipage/index.html>.



Bruno Pereira.

Apresentação.

UFMG, 2014.

<https://copy.com/94BD03mWisUf>.



Bruno Pereira.

Códigos das aplicações apresentadas e extras.

UFMG, 2014.

<https://copy.com/r1XTNWP4H3aM>.



Larry L Peterson and Bruce S Davie.

Computer networks: a systems approach.

Elsevier, 2007.



Jon Postel.

User datagram protocol.

Isi, 1980.

<http://tools.ietf.org/html/rfc768>.



W Richard Stevens.

Tcp slow start, congestion avoidance, fast retransmit, and fast recovery algorithms.

1997.

<http://tools.ietf.org/html/rfc2001>.

Extras

Example

ShowIP.c

- Veja o código ShowIP.c [3]
- Comando: `ping <endereço>`

Example

client_beej.c e server_beej.c

Extras

Example

ShowIP.c

- Veja o código ShowIP.c [3]
- Comando: `ping <endereço>`

Example

client_beej.c e server_beej.c

Extras

Example

ShowIP.c

- Veja o código ShowIP.c [3]
- Comando: `ping <endereço>`

Example

client_beej.c e server_beej.c

- Versão com mais detalhes
- Execute uma instância do servidor e uma do cliente