

# Socket-Aplication

---

Este programa implementa um **socket** em C++.

## Intro - Socket Overview

Sockets são utilizados para comunicação interprocessos, promovendo uma API entre a camada de Aplicação da camada OSI e as inferiores.

Sockets são representados por um *file descriptor* determinados na sua criação, diferentemente de *pipes* podem realizar comunicação suportam comunicações entre processos não relacionados e, mais ainda, são o principal meio de comunicação entre máquinas diferentes. Quando um socket é criado devem-se ser especificados o estilo de comunicação que se deseja usar, o tipo de protocolo que será implementado e o endereço do socket (*namespace*).

- Os estilo de comunicação podem ser:
  1. **SOCK\_STREAM** : Estilo utilizado neste programa.
  2. **SOCK\_DGRAM**
  3. **SOCK\_RAW**

Algumas siglas comuns encontradas são:

*PF\_* : *Protocol Family*

*AF\_* : *Address Family*

Neste programa foi implementado um socket usando os protocolos TCP/IP, sendo o protocolo IP o IPv4.

## Socket.h e outras bibliotecas

Para realização deste socket, além das bibliotecas comuns de C e C++, como *stdio.h*, *stdlib.h*, *strings.h* foi utilizada a biblioteca *sys/socket.h* e suas dependências. Esta biblioteca trata internamente arquiteturas de computadores diferentes no sentido de endereçamento, para isso a classe de bibliotecas de *bits* é incluída. Todas as definições de cada função e estruturas de dados relacionadas utilizadas de *socket* são especificadas no código como comentário.

Para localizar o host através do seu endereço IP foi utilizada a função *gethostbyname()* esta função está em *netdb.h*, por isso esta biblioteca foi incluída.

## TicTacToe : Jogo Da velha

O software descrito neste documento utiliza do mecanismo de sockets para realizar a comunicação entre dois processos que executam o jogo Tic-Tac-Toe (jogo da velha).

## IPv4 Header

O header do protocolo IPv4 é dado da seguinte maneira:

Version(4bits)	...	Protocol(1Bytes)	...	Source IP (4Bytes)	Destination IP(4Bytes)	...
4(IPv4) 0100	...	6(TCP)	...	192.168.0.1	168.172.0.3	...

## Port : Definições de portas

As portas são definidas por 2Bytes, portanto 65536 possibilidades de portas. Contudo existem portas reservadas pela IANA (*Internet Assigned Numbers Authority*) e não é uma boa prática utilizá-las, são elas:

```
0 → 1023 : Reservadas.  
1024 → 49151 : Semi-Reservadas.  
49152 → 65535 : Aplicações em geral.
```

## Usage

A porta que o socket utilizará, bem como o endereço IP da rede local do Server, no caso do *client*, devem ser fornecidos pelo usuário. Como compilar e utilizar este software:

*Host 1(Server):*

```
$make  
$make server  
Enter port number: <port_number>
```

*Host 1(Client):*

```
$make  
$make client  
Enter port number: <port_number>  
Enter IP addr: <ip_addr>
```

*Como descobrir seu IP? - Aqui estão algumas maneiras:*

```
$ifconfig [interface]  
#no meu caso <wlp1s0>  
Link encap:Ethernet  HWaddr 00:00:00:00:00:00  
inet addr:xxx.xxx.xxx.xxx  Bcast:255.255.255.255  Mask:255.255.0.0  
  
$hostname -I  
xxx.xxx.xxx.xxx  
  
$netstat -tpe  
tcp  0  0 xxx.xxx.xxx.xxx:41160  ec2-34-211-99-53.:https ESTABLISHED  
antonio  491685  8683/firefox  
tcp  0  0 xxx.xxx.xxx.xxx:55442  151.101.204.134:https  ESTABLISHED  
antonio  491034  8683/firefox
```

## Fontes

Documentos utilizadas para realização do trabalho (*estão presentes nesta pasta*):

The Linux Programming Interface.pdf  
The GNU C Library Reference Manual.pdf

## Alunos

Antonio Moreira - **9779242**

Leonardo Meireles Murtha Oliveira - **4182085**

Vitor Augusto Henrique Brisola - **9791292**