

# Fundamentos de Comunicação: Sockets e NetCat

Prof. Luiz Lima Jr.

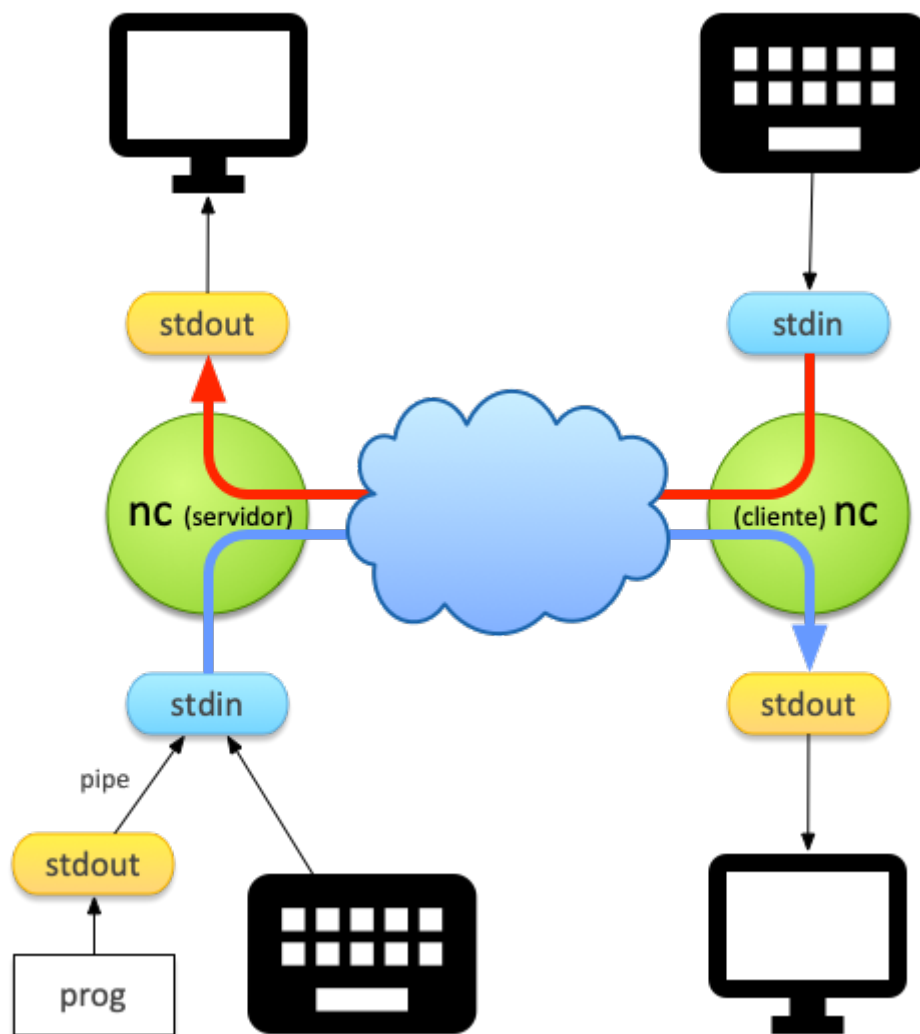
Sistemas Distribuídos e  
Concorrentes  
Programação Distribuída

## NetCat

**NetCat** ou **nc** é um aplicativo do Linux que permite criar e lidar com conexões TCP e UDP. Seus usos mais comuns:

- criação de *proxies*;
- implementação de serviços de rede;
- clientes e servidores HTTP baseados em shell-scripts;
- *daemon* de teste da rede;
- tunelamento de conexões;
- etc.

O NetCat obtém, por **default**, dados da entrada padrão (**stdin**) e envia dados para a saída padrão (**stdout**).

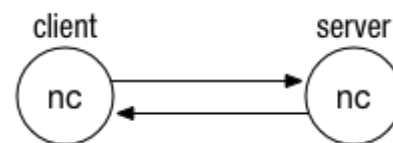


NetCat: estabelece conexão e o que recebe do `stdin`, envia para `stdout`

Veja a **man page** do NetCat (`man nc`) para ver as suas opções e execute as seguintes tarefas (tome notas dos passos necessários em cada caso). Há também disponível uma versão do NetCat para Windows (outras opções no Windows: cygwin ou WSL).

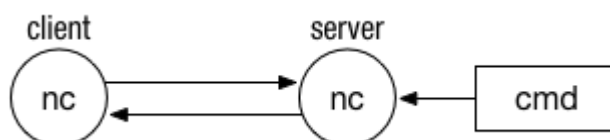
## Chat

1. Crie um servidor TCP que escute em uma porta qualquer ( $> 1024$ );
2. Crie (em outra **shell**) um cliente TCP que se conecte ao servidor. A entrada padrão (`stdin` = teclado) de um processo será enviada ao outro processo.
3. Tente conectar o seu cliente com um servidor criado por um colega seu.
4. Que informações são necessárias para se estabelecer uma conexão entre um cliente e servidor usando sockets?



## Servidor genérico

Na verdade, o NetCat pode ser utilizado para implementar servidores de qualquer tipo. Por exemplo, servidores de arquivos, de listagem de diretórios, de eco, de hora, etc. Basta



conectar a saída de um comando ou aplicação qualquer à entrada do NetCat usando **pipes**.

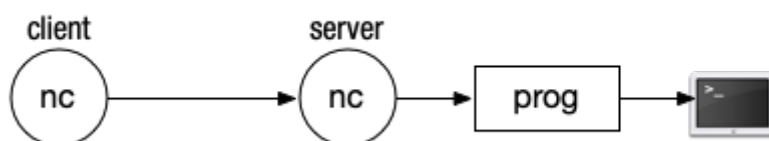
1. O comando "**date**" fornece a data e hora atuais do sistema. Crie um servidor TCP que responde a conexões do cliente, enviando a data e hora do sistema.
2. Crie um cliente que se conecte ao servidor para obter a data-hora atuais.
3. Altere o comando do tópico 1 de forma a fazer o servidor retornar:
  1. a listagem da sua pasta corrente (**ls**);
  2. o conteúdo de um arquivo qualquer (**cat**);
  3. um texto simples qualquer (**echo**);

## Controle Remoto

Crie um programa na sua linguagem de programação preferida que leia os comandos do usuário (**stdin**) e tome as ações correspondentes conforme tabela abaixo:

Comando do Usuário	Ação do programa
<b>hi</b>	Imprime na tela " <b>hello</b> "
<b>shutdown</b>	Termina o programa.

Utilize o NetCat para enviar comandos remotamente ao seu programa.

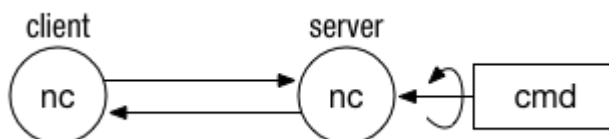


Tente incluir um novo comando **soma** no formato: **soma p1 p2** onde **p1** e **p2** são parâmetros. O servidor deve mostrar na tela o resultado da soma destas parcelas.

## Servidores perenes

O comando da shel:

```
while [ 1 ]; do
<lista_cmds>;
done
```



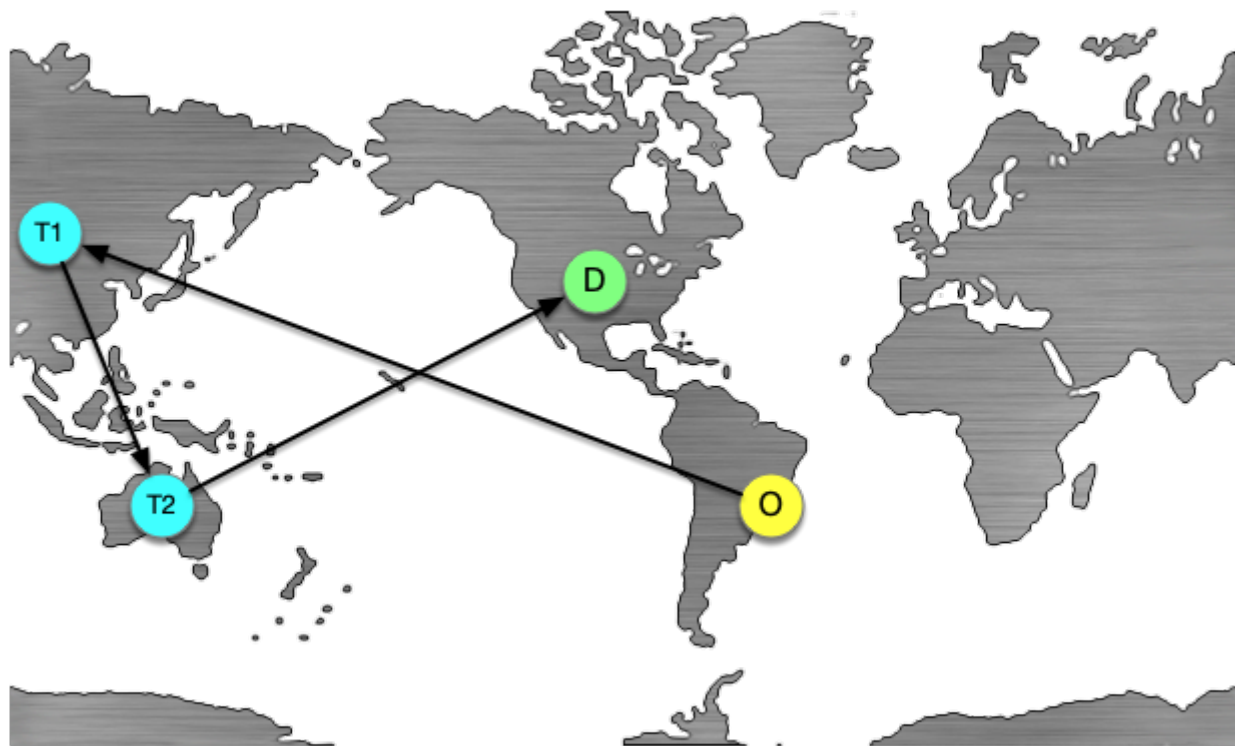
executa em *loop* os comandos especificados em **<lista\_cmds>**.

O comando "**fortune**" produz um texto simples em português na tela.

1. Usando **while**, faça um servidor TCP de textos.
2. Crie um cliente que se conecte ao servidor e obtenha os textos por ele produzidos.

## Redirecionamento de Comunicação

Crie um esquema de "redirecionamento" de comunicação da origem (O) ao destino (D) usando NetCat e **pipes** conforme figura abaixo:



Redirecionamento de mensagem de O para que alcance D passando por T1 e T2

(Utilize o servidor local, alterando apenas as portas.)

## Acessando servidores web

Como utilizar o NetCat para obter uma página web da Internet?

Dicas:

- Servidores web escutam na porta **80** e entendem mensagens do protocolo HTTP (especificação | Tutorial).
- O comando da shell **printf** (**man printf**) permite enviar texto com caracteres especiais: "**\r\n**" (CRLF).

© Luiz A. de P. Lima Jr.