

使用Socket原语概述

实验室练习

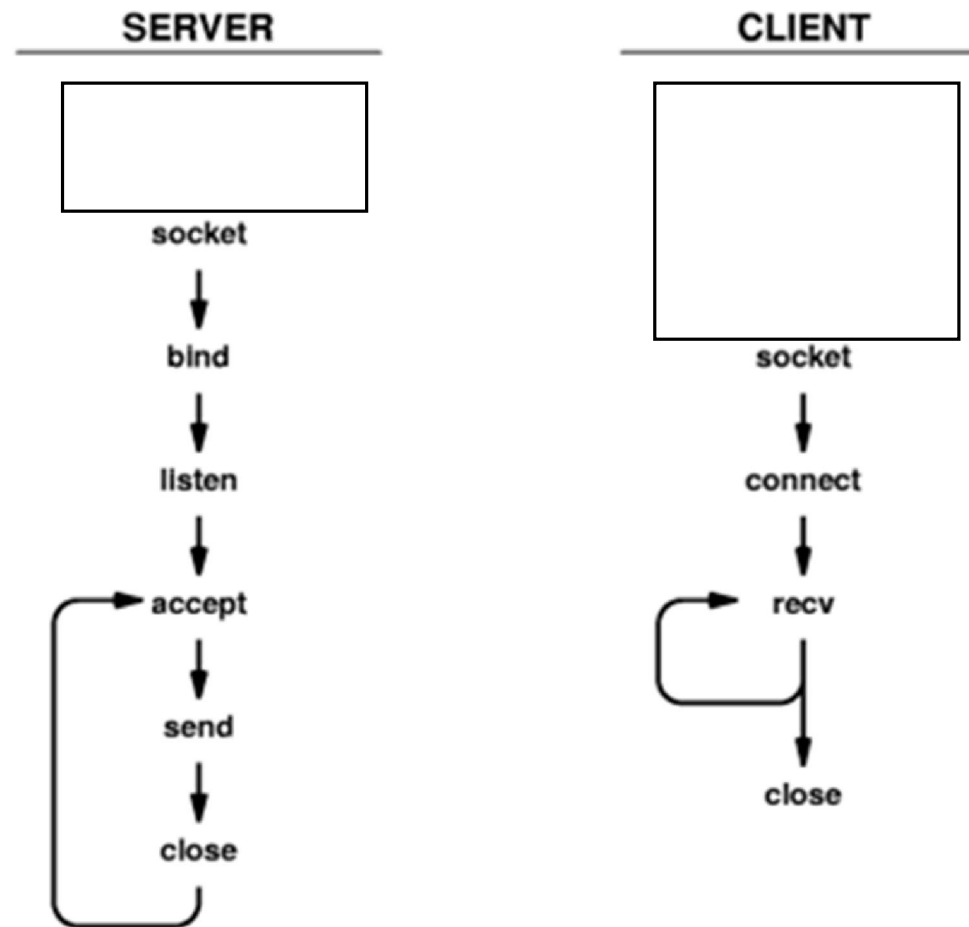
- 请参考前面的类幻灯片上的每个插槽的原语的文字。
- 您以前的实验课参与编写一个众所周知的 *网络应用* 被称为 *白天*。
- 此应用程序的操作如下：
 - 客户机应用程序发出的连接请求，以使用CONNECT原始服务器，
 - 服务器应用程序接受使用接受原始的连接，
 - 服务器应用程序从本地操作系统的日期和时间，并使用SEND原始返回它在一个格式化字符串

通过Socket原语概述

实验室练习

- 服务器应用程序然后关闭使用CLOSE原始的连接，
 - 客户端应用程序使用RECV原始检索来自连接中的数据，其显示在本地 *标准输出* 然后关闭使用CLOSE原语的连接，
 - 客户端应用程序会杀死自己的时候使用内置C-功能 *出口 (0)*；
- 下面的幻灯片突出了原始的调用序列。

示例客户端 - 服务器交互使用 套接字



挑战这周的实验

- 审查了 白天 客户和 服务器 在一些细节上的下一个实验室将建立在这一认识的应用。
- 你必须改变你的 *daytime*客户端 和 服务器 应用程序执行作为 回声 客户和 服务器 应用。
- 该实验室的说明书将概述任务，并给予一定的指针解决问题。

挑战这周的实验

- 此应用程序的要点如下：
 - 客户端应用程序将采取一个字符串作为命令行参数，
 - 这个字符串被发送到服务器通过打开的连接，
 - 服务器应用程序将读取的字符串，并将其返回给客户端应用程序，正是因为它走了进来。
- 要完成这个任务，有必要了解的操作 *的recv ()* 原始。

该的recv () 原始

- 回想其在白天客户端应用程序使用：

```
而 ( ( 的numBytes =的recv ( 袜子 , recvbuffer , BUFSIZE - 1 , 0 ) ) > 0 )
```

```
{  
    recvbuffer [的numBytes] = '\0'; 的fputs ( rec  
    vbuffer , 标准输出 );  
}
```

- 下面的幻灯片概述了这个原始的一些关键点。

该的recv () 原始

- 因为数据可能不以单个呼叫到达while循环是必要的 的recv ()
。
- 的numBytes 从返回值 的recv () :
 - 它代表着从套接字读取的字节数。
 - 它返回三个值中的一个：
 - <1 表示错误状态，
 - 0 表示关闭的连接，并且，
 - > 1 表示具有潜在更多的数据要被接收的开放连接。

挑战这周的实验

- 如果的recv () n个基本在使用上述
回声 客户 和 服务器 应用它会引起问题。
- 该应用程序的任何一个或两者将保持在循环中。
 - 您将需要确定如何解决这个问题。

解决

- 的一个重要方面 *网络应用* 如
白天是 寻址。
- 为了使客户端和服务端应用解决相互沟通某种形式的明确 是 需要。
- 该 *目的地* (记得从本学期 *五元组分通信模型*) 服务器应用程序要求，以便使确定的地址 *资源* 客户端应用程序发起连接请求。

解决

- 回想一下，IP层为互联网的即有利于主机之间传输数据报/包 主机到主机。

- 然而，由于封装这些数据报/包内的数据通常被注定用于

应用 在目标主机上：

- 和可能驻留在应用程序层的许多应用程序之一，
- 寻址的更细的粒度是必需的。
- 这里交通层寻址起着至关重要的作用。

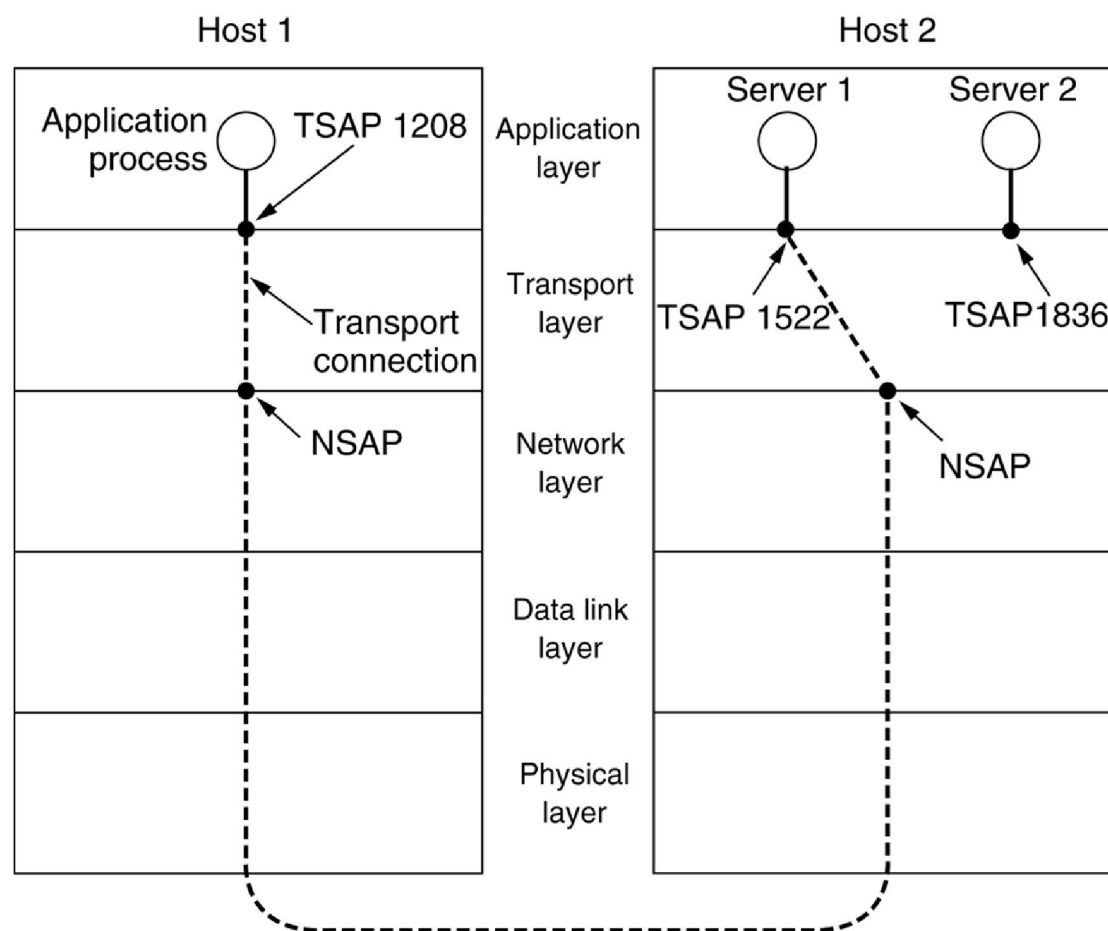
传输寻址

- 回想一下，传输协议应用层提供服务。
- 为了确保明确的个别应用程序的处理，传输层提供了自己的解决方案独立于IP层：
 - 这些通常被称为传输服务访问点 (TSAP时)
 - 这些TSAP可唯一标识称为传输层实体 终点，
 - 在TCP的说法这些端点被称为 端口。

传输寻址

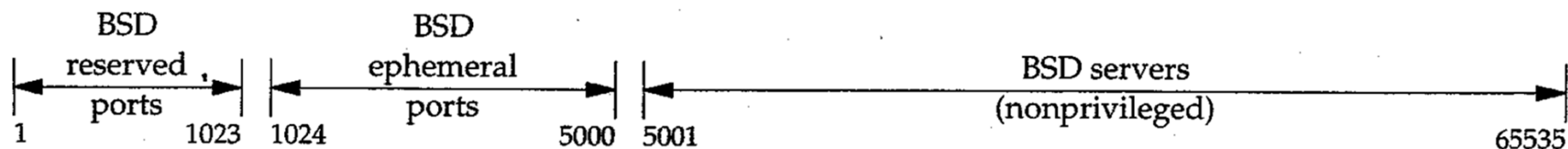
- **港口** 编号使用：
 - 服务器应用到他们的服务做广告，并侦听连接请求，
 - 客户端应用程序进行连接请求时，唯一标识一个服务器应用程序。
- 该 **网络层** 还定义 **终点**。 这些被称为网络服务访问点 (NSAP的)
：
 - IP地址是NSAP的例子。
- 下面滑动示出了NSAP的，和的TSAP传输连接之间的关系

的TSAP，NSAP的运输连接



该 端口号 范围

- TCP端口号是十六位长。
- 这就产生了一个端口号的地址空间包含大约。65K地址（ 16个比特意味着 2^{16} 个地址 ） 如下：



该 端口号 范围

- 保留的地址是公知的应用程序，例如HTTP（端口80），FTP（端口20和21），远程登录（端口23）等。
 - 这些只能由具有SU权限的用户进行分配。
- 短暂的地址由TCP分配给 客户 应用：
 - 这不是明显，在客户端应用程序需要一个端口号，
 - 但是，需要有一个返回地址从服务器应用程序数据。

该 端口号 范围

- 非特权 地址是任何其他应用程序：
 - 这是将在实验练习中使用的范围。
- 值得注意的是，短暂的和非特权范围上的不同操作系统的不同，这一点很重要：
 - 你家里的主机可以使用取决于所使用的操作系统不同的范围。

检查寻址的详细信息

- 的操作 端口号 从TCP透视可以使用被看作 *netstat*的

效用：

- 只需输入命令，*netstat*的-*ntap* 在命令行提示
- 此命令揭示了存在于主机操作系统内的连接的详细信息。

检查寻址的详细信息

- 标志的解释：
 - “N”揭示了点分十进制形式的IP地址，
 - 在TCP“T”仅过滤地址，
 - “A”显示所有连接，
 - “P”揭示了与每个连接相关联的应用。
- 下面幻灯片显示了一个示例输出，当 **netstat** 的命令被使用。

检查寻址的详细信息

原的Recv-Q发送-Q本地地址				国外地址	州	PID / 节目名称
TCP	0	0	0.0.0.0:1022	0.0.0.0:*	听	12937 / Web服务器
TCP	0	0	147.252.30.9:1022	147.252.234.34:4136	ESTABLISHED	13268 / Web服务器
TCP	0	0	147.252.234.34:4136	147.252.30.9:1022	ESTABLISHED	13267 / HttpClient的

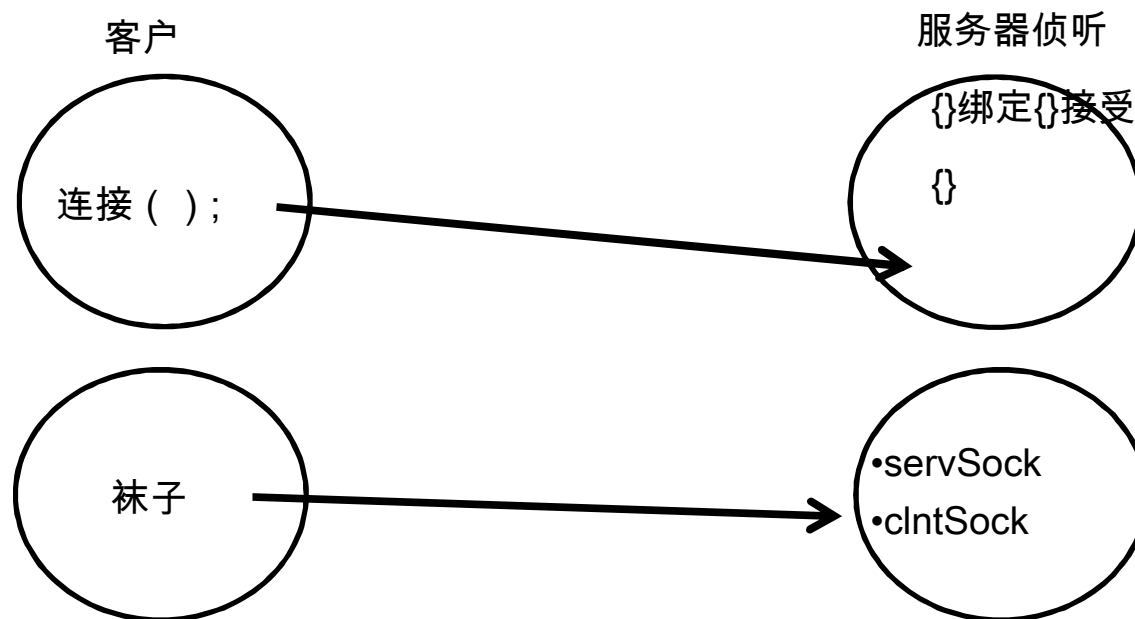
- 由此可见：
 - 与服务器 *听* 和 *连接的* 在端口1022插座，
 - 在一个客户端 *短暂的* 端口4136。

双插座

- 注意列本地地址和远程地址：
 - 这指的是在的TSAP 每 一个连接的端部，
 - 这取决于你是从查看它的连接的哪一端不同的秩序。
- 本地地址和远程地址的这种组合称为 — **插口对**：
 - {147.252.30.9:1022 , 147.252.234.34:4136}是怎么连接从服务器的TCP角度来看。

双插座

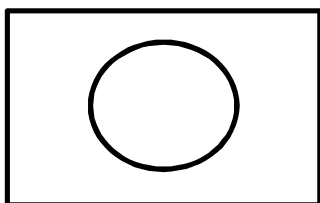
- 下面的幻灯片揭示了如何连接从一个插口对观点成立：
 - 它假定以下基本已吁请导致产生客户端和服务端应用程序中的插座标识符：



之前和连接建立后，套接字

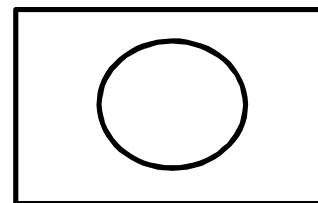
之前 连接建立

198.69.10.2



客户端套接字：{198.69.10.2 0.1500}

206.62.226.35



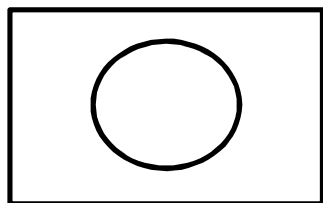
服务器监听套接字：{* 0.21, *. *}

客户端连接请求：206.62.226.35，端口21



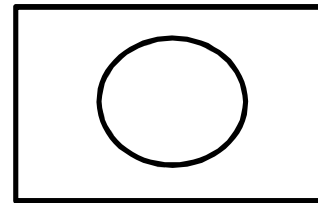
后 连接建立

198.69.10.2



客户端连接套接字：{198.69.10.2 0.1500, 206.62.226.35.21}

206.62.226.35



服务器连接套接字：{* 0.21, *. *}
服务器连接套接字：{206.62.226.35.21, 198.69.10.2.1500}