

UC Day15

预习课

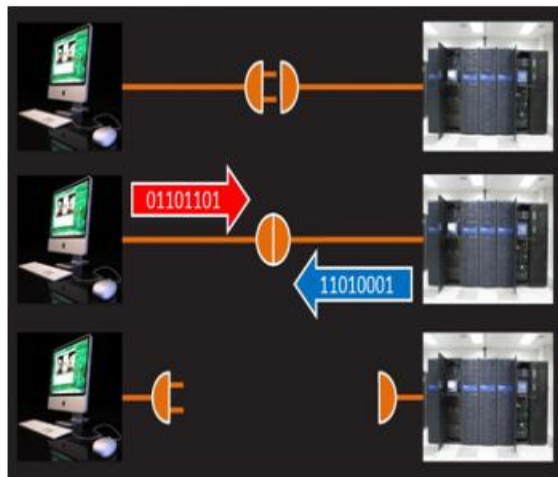
预习
内容

TCP协议

TCP协议

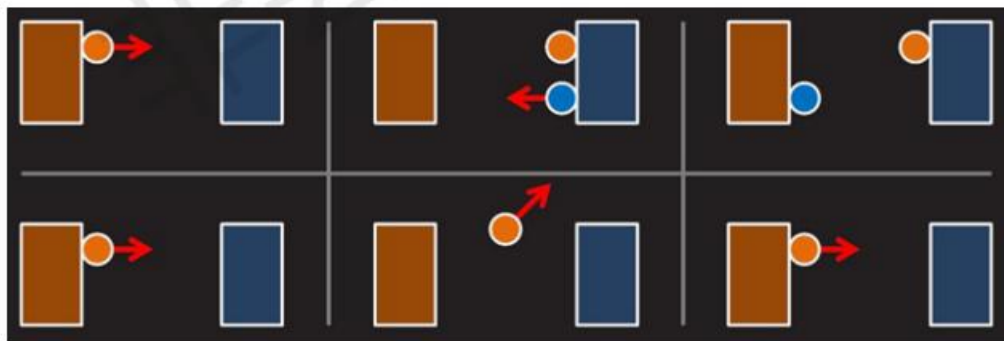
TCP协议

- TCP提供客户机与服务器的连接
 - 一个完整TCP通信过程需要依次经历三个阶段
 - 首先，客户机必须建立与服务器的连接，所谓虚电路
 - 然后，凭借已建立好的连接，通信双方相互交换数据
 - 最后，客户机与服务器双双终止连接，结束通信过程



TCP协议

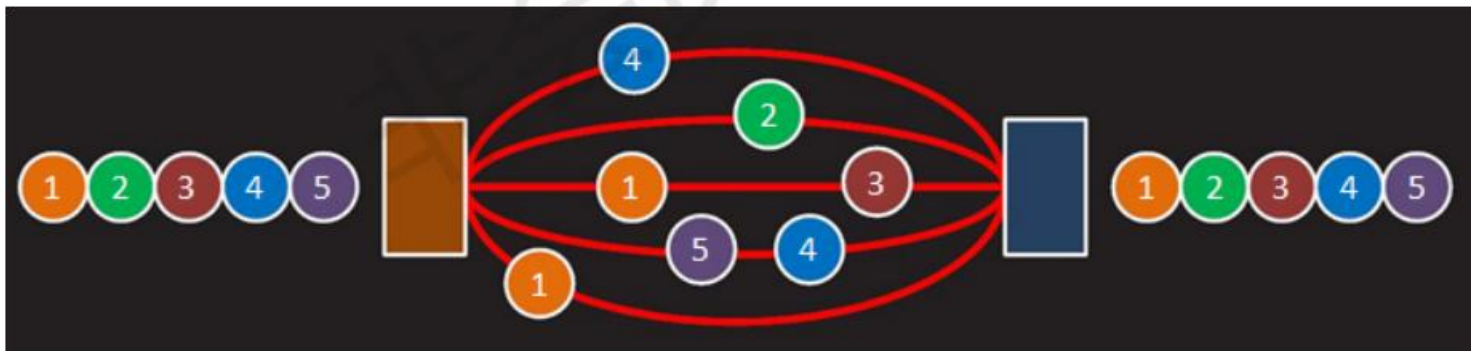
- TCP保证数据传输的可靠性
 - TCP的协议栈底层在向另一端发送数据时，会要求对方在一个给定的时间窗口内返回确认。如果超过了这个时间窗口仍没有收到确认，则TCP会重传数据并等待更长的时间。只有在数次重传均告失败以后，TCP才会最终放弃。TCP含有用于动态估算数据往返时间(Round-Trip Time, RTT)的算法，因此它知道等待一个确认需要多长时间



TCP协议

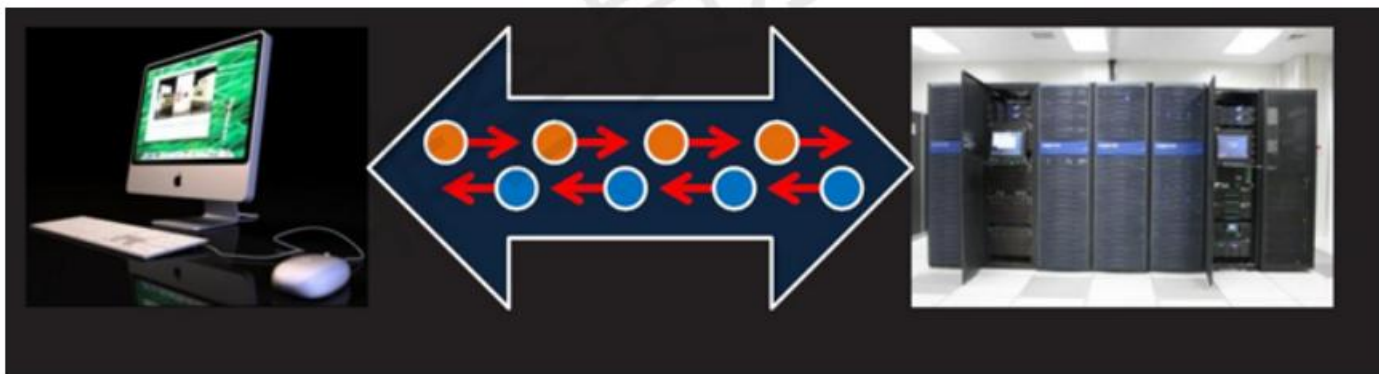
- TCP保证数据传输的有序性

- TCP的协议栈底层在向另一端发送数据时，会为所发送数据的每个字节指定一个序列号。即使这些数据字节没有能够按照发送时的顺序到达接收方，接收方的TCP也可以根据它们的序列号重新排序，再把最后的结果交给应用程序



TCP协议

- TCP是全双工的
 - 在给定的连接上，应用程序在任何时候都既可以发送数据也可以接收数据。因此，TCP必须跟踪每个方向上数据流的状态信息，如序列号和通告窗口的大小



TCP协议

- 三路握手

- 客户机的TCP协议栈向服务器发送一个SYN分节，告知对方自己将在连接中发送数据的初始序列号，谓之主动打开
- 服务器的TCP协议栈向客户机发送一个单个分节，其中不仅包括对客户机SYN分节的ACK应答，还包含服务器自己的SYN分节，以告知对方自己在同一连接中发送数据的初始序列号
- 客户机的TCP协议栈向服务器返回ACK应答，以表示对服务器所发SYN的确认



TCP协议

- 三路握手



TCP协议

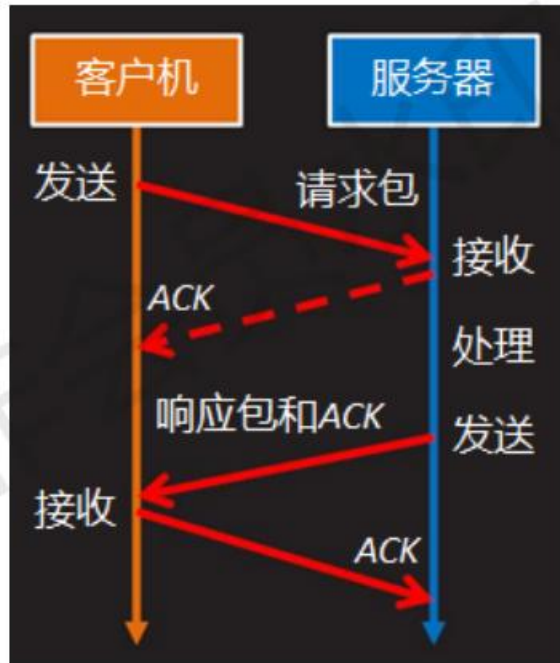
- 交换数据

- 一旦连接建立，客户机即可构造请求包并发往服务器 服务器接收并处理来自客户机的请求包，构造响应包
- 服务器向客户机发送响应包，同时捎带对客户机请求包的ACK应答
- 客户机接收来自服务器的响应包，同时向对方发送ACK应答



TCP协议

- 交换数据



TCP协议

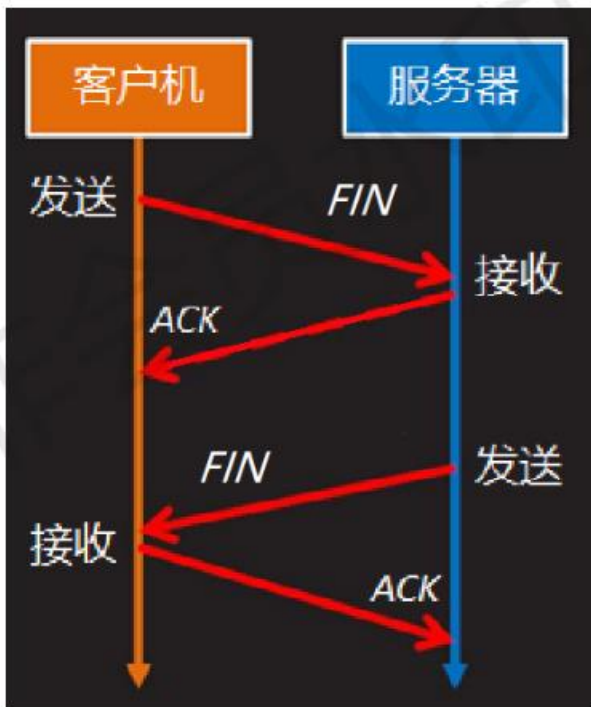
- 四次挥手

- 客户机或者服务器主动关闭连接，TCP协议栈向对方发送FIN分节，表示数据通信结束。如果此时尚有数据滞留于发送缓冲区中，则FIN分节跟在所有未发送数据之后
- 接收到FIN分节的另一端执行被动关闭，一方面通过TCP协议栈向对方发送ACK应答，另一方面向应用程序传递文件结束符
- 一段时间以后，方才接收到FIN分节的进程关闭自己的连接，同时通过TCP协议栈向对方发送FIN分节
- 对方在收到FIN分节后发送ACK应答



TCP协议

- 四次挥手



直播课见

UC

C/C++教学体系

目录

TCP函数

编程模型

通信终止

TCP函数

TCP函数

- `#include <sys/socket.h>`
- `int listen(int sockfd, int backlog);`
 - 功能：启动侦听：在指定套接字上启动对连接请求的侦听功能
 - 参数：sockfd：套接字描述符，在调用此函数之前是一个主动套接字，是不能感知连接请求的，在调用此函数并成功返回之后，是一个被动套接字，具有感知连接请求的能力
 - backlog：未决连接请求队列的最大长度，一般取不小于1024的值
 - 返回值：成功返回0，失败返回-1



TCP函数

- `#include <sys/socket.h>`
- `int accept(int sockfd, struct sockaddr* addr, socklen_t* addrlen);`
 - 功能：等待并接受连接请求，在指定套接字上阻塞，直到连接建立完成
 - 参数：
sockfd：侦听套接字描述符
addr：输出连接请求发起方的地址信息
addrlen：输出连接请求发起方的地址信息字节数
 - 返回值：成功返回可用于后续通信的连接套接字描述符，失败返回-1



TCP函数

- `#include <sys/socket.h>`
- `ssize_t recv(int sockfd, void* buf, size_t count, int flags);`
 - 功能：接收数据
 - 参数：若flags取0则与read函数完全等价，另外也可取以下值
 - `MSG_DONTWAIT` - 以非阻塞方式接收数据
 - `MSG_OOB` - 接收带外数据
 - `MSG_WAITALL` - 等待所有数据，即不接收到count字节就不返回
 - 返回值：成功返回实际接收到的字节数，失败返回-1



TCP函数

- `#include <sys/socket.h>`
- `ssize_t send(int sockfd, void const* buf, size_t count, int flags);`
 - 功能：发送数据
 - 参数：若flags取0则与write函数完全等价，另外也可取以下值
 - `MSG_DONTWAIT` - 以非阻塞方式接收数据
 - `MSG_OOB` - 接收带外数据
 - `MSG_DONTROUTE` - 不查路由表，直接在本地网络中寻找目的主机
 - 返回值：返回值：成功返回实际发送的字节数，失败返回-1



编程模型

TCP编程模型

- TCP编程模型

步骤	服务器		客户机		步骤
1	创建套接字	socket	socket	创建套接字	1
2	准备地址结构	sockaddr_in	sockaddr_in	准备地址结构	2
3	绑定地址	bind	——	——	——
4	启动侦听	listen	——	——	——
5	等待连接	accept	connect	请求连接	3
6	接收请求	recv	send	发送请求	4
7	发送响应	send	recv	接收响应	5
8	关闭套接字	close	close	关闭套接字	6



通信终止

通信终止

- 客户机主动终止通信过程

- 在某个特定的时刻，客户机认为已经不再需要服务器继续为其提供服务了。于是它在接收完最后一个响应包以后，通过close函数关闭与服务器通信的套接字。客户机的TCP协议栈向服务器发送FIN分节并得到对方的ACK应答。服务器专门负责与该客户机通信的子进程，此刻正试图通过recv函数接收下一个请求包，结果却因为收到来自客户机的FIN分节而返回0。于是该子进程退出收发循环，同时通过close函数关闭连接套接字，导致服务器的TCP协议栈向客户机发送FIN分节，使对方进入TIME_WAIT状态，并在收到对方的ACK应答以后，自己进入CLOSED状态。随之收发循环的退出，服务器子进程终止，并在服务器主进程的SIGCHLD(17)信号处理函数中被回收。通信过程宣告结束



通信终止

- 服务器主动终止通信

- 服务器专门负责和某个特定客户机通信的子进程，在运行过程中出现错误，不得不调用close函数关闭连接套接字，或者直接退出，甚至被信号杀死。于是服务器的TCP协议栈向客户机发送FIN分节并得到对方的ACK应答
- A、如果客户机这时正试图通过recv函数接收响应包，那么该函数会返回0。客户机可据此判断服务器已宕机，直接通过close库函数关闭与服务器通信的套接字，终止通信过程
- B、如果客户机这时正试图通过send函数发送请求包，那么该函数并不会失败，但会导致对方以RST分节做出响应，该响应分节甚至会先于FIN分节被紧随其后的recv函数收到并返回-1，同时置errno为ECONNRESET。这也是终止通信的条件之一



通信终止

- 服务器主机不可达(主机崩溃、网络中断、路由失效等)
 - 在服务器主机不可达的情况下，无论是客户机还是服务器，它们的TCP协议栈都不可能再有任何数据分节的交换。因此，客户机通过send函数发送完请求包以后，会阻塞在recv函数上等待来自服务器的响应包。这时客户机的TCP协议栈会持续地重传数据分节，试图得到对方的ACK应答。源自伯克利的实现最多重传12次，最长等待9分钟。当TCP最终决定放弃时，会通过recv函数向用户进程返回失败，并置errno为ETIMEOUT或EHOSTUNREACH或ENETUNREACH。在重传过程中会放弃以后，即使服务器主机被重启，或者通信线路被恢复，由于TCP协议栈依然丢失了先前与连接有关的信息，通信依然无法继续，对所接收到的一切数据一律响应RST分节，只有在重新建立TCP连接之后，才能继续通信



谢谢

UC Day015

复习课

域名解析

域名解析

- IP地址是网络上标识站点的数字地址，为了方便记忆，采用域名来代替IP地址标识站点地址
- 域名解析就是域名到IP地址的转换过程。域名的解析工作由DNS服务器完成
- 当应用过程需要将一个主机域名映射为IP地址时，就调用域名解析函数，解析函数将待转换的域名放在DNS请求中，以UDP报文方式发给本地域名服务器。本地的域名服务器查到域名后，将对应的IP地址放在应答报文中返回



域名解析

- `#include <netdb.h>`
- `struct hostent* gethostbyname(char const* host_name);`
 - 功能：通过参数所传的主机域名，获取主机信息
 - 参数：host_name 主机域名
 - 返回值：函数执行成功返回表示主机信息的结构体指针，失败返回NULL
 - 注意，该函数需要再联网情况下使用



域名解析

- struct hostent {
 char *h_name; //主机官方名
 char **h_aliases; // 主机别名表
 int h_addrtype; //地址类型
 int h_length; //地址长度
 char **h_addr_list; //IP地址表
};



域名解析

- 对于WEB服务器而言，主机官方名有一个，而主机别名可能有多，这些别名都是为了便于用户记忆。同时IP地址也可能有多。

```
- h_aliases -> * -> "xxx\n"
               * -> "xxx\n"
               NULL;

- h_addr_list -> * -> in_addr
                  * -> in_addr
                  NULL
```



下节课见