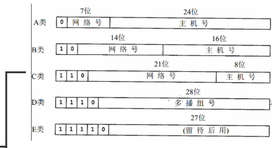
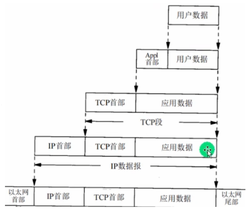
网络知识

- IP地址
  查看本机ip: ipconfig
  - 唯一标识网络中每一台计算机
  - 分支主题 2
  - IPv4
    - ip地址为网络地址+主机地址



类型	范 围
A	0.0.0.0 到 127.255.255.255
B	128.0.0.0 到 191.255.255.255
C	192.0.0.0 到 223.255.255.255
D	224.0.0.0 到 239.255.255.255
E	240.0.0.0 到 247.255.255.255

    - IPv6 — 16字节 如: — 分支主题 1
- 端口
  - 域名+端口可以找到同一台主机对应的不同网络服务
  - 多个服务不能使用同一端口
  - 范围: 2个字节: 65535
    0~1024已经被占用如: http等
  - **注意: 客户端和服务器的某一端口交互时所用的端口是系统随机分配的不能确定**
- 域名
- 协议



- netstat指令

  分支主题 1

  这里的established表示
  本地地址的某一个端口和外部地址的某一个端口建立了连接

java网络编程

- InetAddress
  - 获取InetAddress对象
    - //获取本机InetAddress对象(本机主机名+IP地址)
      InetAddress inetAddress = InetAddress.getLocalHost();
    - 根据主机名(本机主机名+IP地址)
      InetAddress inetAddress1 = InetAddress.getByName(主机名称/主机IP地址);
    - //根据域名获取对象
      InetAddress inetAddress2 = InetAddress.getByName("www.baidu.com");
    - 概括 1
  - 根据InetAddress对象获取信息 — //根据对象获取IP地址和域名
    System.out.println(inetAddress2.getHostAddress()+" " +inetAddress2.getHostName());
- Socket
  - TCP
    - 服务器端
      - 字节流 — 调用StreamUtils — 调用StreamUtils
        - 将客户端上传的文件/文件路径保存到客户端/数据库
          - 创建Socket实例对象 — 服务器端: ServerSocket serverSocket = new ServerSocket(端口号);
            每一个客户端连接都会创建一个Socket对象,
            Socket accept = serverSocket.accept();
          - //读取客户端发来的文件
            BufferedInputStream bufferedInputStream = new BufferedInputStream(accept.getInputStream());
            byte[] buffer = StreamUtils.copyToByteArray(bufferedInputStream);
            accept.shutdownInput();
          - //保存文件到服务的
            BufferedOutputStream bufferedOutputStream = new BufferedOutputStream(new FileOutputStream("hello.jpg"));
            bufferedOutputStream.write(buffer);
            accept.shutdownOutput();
          - 关闭外层流后,内层流也会自动关闭
            serverSocket.close();
            bufferedInputStream.close();
            bufferedOutputStream.close();
        - 将服务器上的文件传输给客户端
      - 最后要关闭所有创建的流
      - 字符流 — **和字节流差不多,但记得得用char数组**
    - 客户端
      - 字节流 — 调用StreamUtils
        - 上传客户端文件到服务器
          - 创建Socket实例对象 — 客户端: Socket socket = new Socket(连接的主机信息(服务器的InetAddress对象),端口号);
          - //读取客户端文件
            BufferedInputStream bufferedInputStream = new BufferedInputStream(new FileInputStream("D:\\OneDrive\\桌面\\fs\\photo\\fs.jpg"));
            byte[] buffer = StreamUtils.copyToByteArray(bufferedInputStream);
            socket.shutdownInput();
            System.out.println("读取完成");
          - //上传服务器
            BufferedOutputStream bufferedOutputStream = new BufferedOutputStream(socket.getOutputStream());
            bufferedOutputStream.write(buffer);
            socket.shutdownOutput();
            System.out.println("上传完成");
          - //关流
            socket.close();
            bufferedInputStream.close();
            bufferedOutputStream.close();
        - 从服务器下载文件到客户端
      - 最后要关闭所有创建的流
      - 字符流 — **字节流差不多,记得用char数组**
  - UDP
    - 说明 — 可一次性最多发送的数据报,长度为64k byte
    - 双方
      - 发送方
        - //构建DatagramSocket对象
          DatagramSocket datagramSocket = new DatagramSocket(9998);
        - //发送数据
          byte[] bytes = "hello world".getBytes();
          DatagramPacket datagramPacket = new DatagramPacket(bytes(发送数据转换为字节数组后的数组名),bytes.length(数组长度), InetAddress.getByName("192.168.1.8")(接收方InetAddress对象),9999(接收方端口));
          datagramSocket.send(datagramPacket);
        - //关流
          datagramSocket.close();
      - 接收方
        - 1.DatagramSocket datagramSocket = new DatagramSocket(端口号);
        - //接收datagramPacket数据报,这里的数据长度最多为64k 即 1024*64
          byte[] buffer = new byte[1024];
          DatagramPacket datagramPacket = new DatagramPacket(buffer(数组名),buffer.length(数组长度));
          //如果没有数据到来,会阻塞,直到数据到来.
          System.out.println("等待数据接收中~~~");
          //接收数据
          datagramSocket.receive(datagramPacket);
        - 获取接收到的数据的属性
          byte[] bytes = datagramPacket.getData();
          int len = datagramPacket.getLength();
          System.out.println("接收到的数据是: " + new String(bytes,0,len) + "\n" + "字节长度为" + len);
        - //关流
          datagramSocket.close();

注意: 只要是读取或发送到对方的流一定是soket.getInputStream()或socket.getOutputStream();创建出来的

获取InetAddress对象

获取InetAddress对象