Chương 5: LẬP TRÌNH MẠNG

5.1. Nền tảng giao tiếp

Câu 1.1: Tạo sao chức năng hoạt động mạng là yêu cầu cơ bản của hầu hết thiết bị di động? Câu 1.1: Kết nối mạng của thiết bị di động bao gồm các loại mạng nào?

- Mạng viễn thông lẫn mạng internet.

Câu 1.2: Chức năng hoạt động mạng thường được thiết kế sẵn bởi nhà sản xuất thiết bị di động, tại sao cần lập trình mạng? Ví dụ?

- Trong một số trường hợp người phát triển phần mềm cần lập trình mạng cho ứng dụng của mình.

Câu 1.3: Android hỗ trợ hầu hết các loại kết nối mạng nào?

- Hầu hết các loại kết nối mạng như: GSM, GPRS (2G, 3G, 4G), Bluetooth, Wifi, GPS.

Câu 2: Trình bày các bước lập trình để kiểm tra xem điện thoại đang sử dụng loại kết nối nào?

(1) Tạo ra đối tượng **ConnectivityManager** với lớp ConnectivityManager , cú pháp:

```
ConnectivityManager cm =
(ConnectivityManager)getSystemService(Context.CONNECTIVITY_SERVICE);
```

(2) Lấy **biến** thông tin về mạng với hàm **getAtiveNetworkInfo(),** cú pháp:

```
NetworkInfo info = cm.getActiveNetworkInfo();
```

(3) Lấy thông tin về loại mạng đang kết nối với hàm getType(), cú pháp:

```
int type = info.getType();
```

(4) Viết các lệnh hiển thị thông thông kết nối mạng

5.2. Lướt web với Android

Câu 3: Với Android, ta có thể nhúng trình duyệt web có sẵn dưới dạng một widget vào trong các Activity để hiển thị các nội dung HTML hoặc để duyệt Internet. Android browser dựa trên lớp nào? Và dùng widget nào để làm chỗ trú cho các trang của trình duyệt?

- Android browser dựa trên WebKit và dùng widget **WebView** để làm chỗ trú cho các trang của trình duyệt.

Cầu 4: Để Activity truy nhập được Internet và tải các trang web vào một WebView, ta phải bổ sung permission gì và vào đâu?

- Ta phải bổ sung INTERNET permission vào tập tin Android Manifest, cú pháp:

<uses-permission android:name="android.permission.INTERNET" />

Câu 5: Trong tập tin Layout XML ta phải khai báo đối tượng nào?

- WebView

Câu 6: Phương thức được dùng để nhập URL của trang web là gì?

- loadUrl("…")

Câu 7: Để đọc được Javascript trên trang web, ta gọi phương thức nào? Cú pháp?

- browser.setSettings().**setJavaScriptEnabled**(true); // browser là một WebView.

5.3. Telephony API

Câu 8: Để gửi một tin nhắn SMS, ta sử dụng đối tượng nào? Chức năng? Cách lấy đối tượng này?

- **SmsManager**, quản lý việc vận hành SMS như data, text, và pdu SMS messages, để lấy đối tượng này ta gọi phương thức **getDefault()**.

Câu 9: Phương thức sendDataMessage được dùng khi nào? Các đối số không null.

- Gởi dữ liệu bằng tin nhắn SMS

- **public void sendDataMessage** (String destinationAddress, String scAddress, short destinationPort, byte[] data, PendingIntent sentIntent, PendingIntent deliveryIntent)

- destinationAddress: địa chỉ để gởi message tới.
- destinationPort: port để gởi message tới.
- byte[] data: dữ liệu

Câu 10: Phương thức sendTextMessage được dùng khi nào? Các đối số không null.

- Gởt text bằng tin nhắn SMS

- **public void sendTextMessage** (**String destinationAddress, String scAddress, String text**, PendingIntent sentIntent, PendingIntent deliveryIntent).

5.3. Telephony API (tiếp theo)

Câu 11: Để thực hiện gởi tin nhắn SMS giữa 2 điện thoại ta phải khai báo permission gì vào tập tin AndroidManifest.xml ?

- *<uses-permission android:name="android.permission.SEND_SMS"/>*
- *<uses-permission android:name="android.permission.RECEIVE_SMS"/>*

Câu 12: Để xây dựng chương trình để nhận tin nhắn, chương trình chỉ bao gồm một danh sách hiển thị tất cả tin nhắn trong Inbox, ta sử dụng lớp nào?

- ListActivity

Câu 12: Lệnh Cursor c = getContentResolver().query(SMS_INBOX, null, null, null, null); thực hiện việc gì?

- Tạo ra một đối tượng con trỏ dùng để đọc tin nhắn trong thư mục inbox.

Câu 13: Lệnh startManagingCursor (c), với c là đối tượng con trỏ, thực hiện việc gì?

- Báo cho Activity hiện tại quản lý vòng đời của con trỏ vừa mới tạo ra

Câu 13: Để nhận và đọc tin nhắn ta phải bổ sung permission gì vào tập tin AndroidManifest.xml?

- <uses-permission android:name="android.permission.READ_SMS"></uses-permission>

5.3. Telephony API (tiếp theo)

Câu 14: Một tình huống thường xảy ra là khi đang chạy 1 ứng dụng trên điện thoại (như nghe nhạc hay xem phim) thì có cuộc gọi đến. Khi đó ta cần bắt được các sự kiện khi có cuộc gọi để chức năng đang sử dụng tạm dừng. Hoặc khi cuộc điện thoại kết thúc ta cần tiếp tục chương trình. Để làm được việc này ta có thể sử dụng lớp nào? Và khai báo như thế nào?

- TelephonyManager
- TelephonyManager teleMgr = (TelephonyManager)**getSystemService**(Context.TELEPHONY_SERVICE);

Câu 15: Tại sao các điện thoại đều hỗ trợ chức năng gởi/nhận tin nhắn SMS, MMS và email, nhưng ta cũng phải biết lập trình cho việc này?

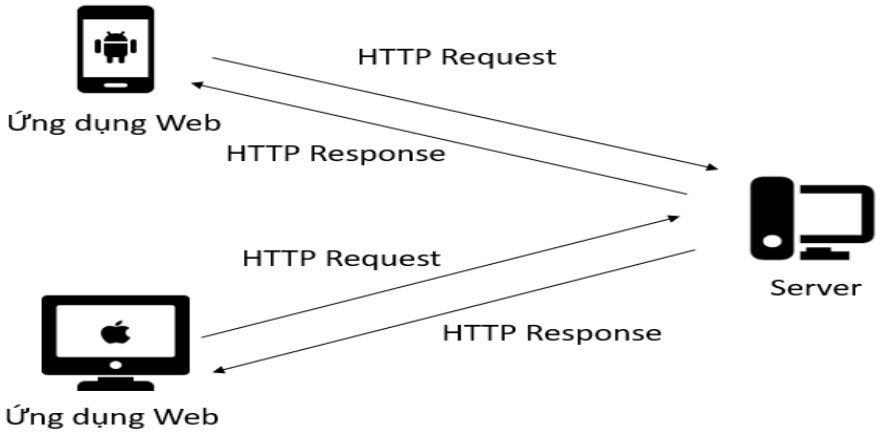
- Vì trong một số ứng dụng cần thực hiện việc này, ta phải lập trình.

Câu 16: Ta có thể cung cấp trực tiếp mã HTML để trình duyệt hiển thị hay không? Ví dụ?.

- Ta có thể cung cấp thẳng mã HTML để trình duyệt hiển thị (chẳng hạn một hướng dẫn sử dụng).
- Ví dụ: browser.loadData("<html><body>Hello, world!</body></html>", "text/html", "UTF-8"); (với browser là một WebView)

5 .4. HTTP (Hypertext Transfer Protocol)

Câu 17: Giao thức HTTP hoạt động như thế nào?



HTTP là một giao thức giao tiếp trên cơ sở của TCP/IP, là một giao thức dạng yêu cầu và đáp ứng.

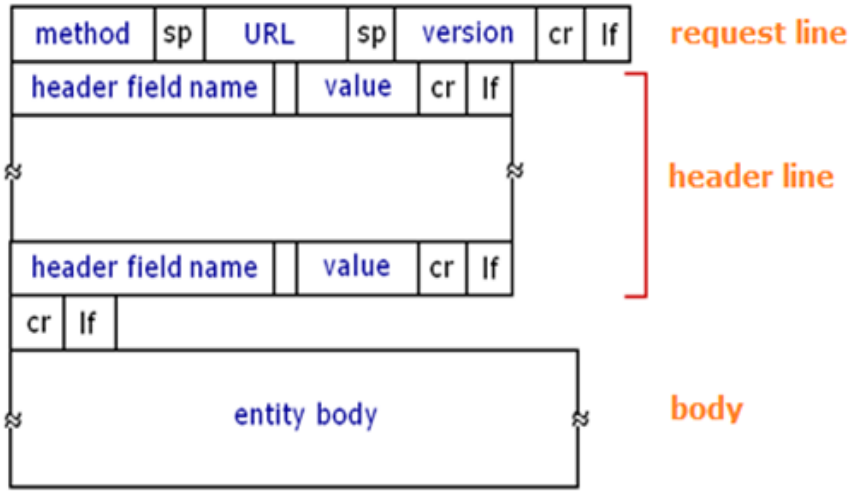
Theo đó, trình duyệt web khởi tạo một yêu cầu (request) rồi gởi đến máy chủ web có địa chỉ được chỉ ra trong URL. Sau khi tiếp nhận yêu cầu, máy chủ web có nvụ xử lý và gởi đáp ứng (response) ngược trở lại cho trình duyệt. Quá trình cụ thể như sau: Người dùng nhập URL (ví dụ: http://www.cit.ctu.edu.vn) vào thanh địa chỉ của trình duyệt trên máy khách (client); trình duyệt đóng gói thông tin yêu cầu; gởi qua mạng; máy chủ (server) nhận xử lý và gởi đáp ứng; máy khách nhận thông tin đáp ứng và hiển thị cho người dùng.

5.4. HTTP (tt)

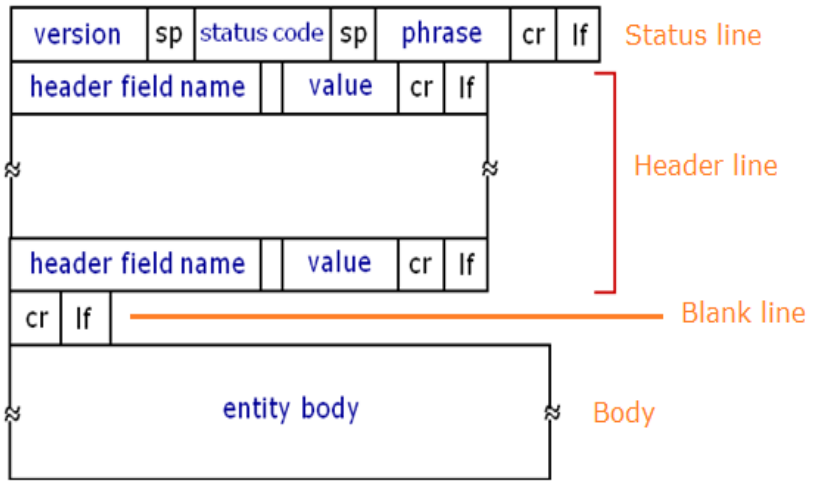
Câu 18: Client gởi một Request đến Server để yêu cầu thực thi một tác vụ hoặc lấy một dữ liệu nào đó. Đối tượng Request chứa các tham số nào?

- URL gởi đến cùng với các tham số khác để xác định tác vụ cần thực thi cũng như loại dữ liệu cần lấy. Có 2 dạng thông điệp HTTP: Thông điệp yêu cầu (HTTP request) và thông điệp đáp ứng (HTTP respone)

HTTP request



HTTP response



sp: các giá trị về khoảng trống.

Blank line: bao gồm các giá trị điều khiển trở về đầu dòng (cr), xuống hàng (lf),

status code: mã trạng thái, phrase: trạng thái tương ứng .

Entity Body (nếu có): là phần thân của thông điệp HTTP yêu cầu.

5.4. HTTP (tt)

Câu 19: Cho biết các mã trạng thái (Static code) trong header của thông điệp đáp ứng?

Mã	Miêu tả
1xx: Thông tin	Yêu cầu đã được nhận và tiến trình đang tiếp tục.
2xx: Thành công	Hoạt động đã được nhận, được hiểu, và được chấp nhận một cách thành công.
3xx: Sự điều hướng lại	Hoạt động phải được thực hiện để hoàn thành yêu cầu.
4xx: Lỗi Client	Yêu cầu chứa cú pháp không chính xác hoặc không được thực hiện.
5xx: Lỗi Server	Server thất bại với việc thực hiện một yêu cầu.

5.4. HTTP (tt)

Câu 20: Có các loại request nào? Các loại này khác nhau như thế nào?

- GET, POST, HEAD, PUT, DELETE,...

- **Yêu cầu GET** (GET request): yêu cầu thông tin - dữ liệu được chuyển đi như là một phần của URL, được dùng để lấy dữ liệu từ server và đây là phương thức mặc định.
- **Yêu cầu POST:** yêu cầu thông tin - dữ liệu được chuyển đi trong luồng riêng biệt (không nằm trong URL), có thể dùng để gởi dữ liệu đến server.
- **Yêu cầu HEAD:** yêu cầu thông tin về thông tin (meta-information), tương tự như GET nhưng không có dữ liệu trả về từ server. Nó có thể dùng để kiểm tra tính hợp lệ của một địa chỉ URL.

Câu 21: Nêu lên các ưu và nhược điểm của GET so với POST

- GET thực thi nhanh hơn POST vì những dữ liệu gửi đi luôn được lưu lại trong cache của trình duyệt. Khi gửi dữ liệu đi trình duyệt sẽ xem trong cache có kết quả tương ứng với yêu cầu đó không và trả về ngay không cần phải thực thi các yêu cầu đó ở phía Server.
- GET không bảo mật so với loại POST
- GET còn bị giới hạn số ký tự của URL của trình duyệt.

5.4. HTTP (tt)

Câu 22: Phân tích URL sau:

http://www.cit.ctu.edu.vn/index.php?option=com_content&task=view&id=2290&Itemid=368&lang=vi

- Cho biết kết nối trong chương trình này là Http GET. Thông tin dữ liệu được chuyển đi là một phần của url, tất cả thông tin được gởi qua cặp "khóa – giá trị" (ví dụ: option=com_content, task=view). Ký hiệu "?": phân cách phần địa chỉ URL và thông tin dữ liệu, ký hiệu "&" phân cách các cặp khóa-giá trị.

Câu 23: Trong URL sau: http://localhost:8080/myApp?first=joe&last=cool:

- http là giao thức; localhost là host; 8080 là port ; /myApp là tập tin ; first=joe&last=cool là các cặp « khóa-giá trị » của yêu cầu.

Câu 24: Một kết nối HTTP có thể ở một trong các trạng thái nào?:

- Thiết lập (Setup), Kết nối (Connect) và Đóng (Close).

5.4. HTTP (tt)

Câu 25: Cho nhận xét về quá trình tạo một kết nối HTTP?

- Trong trạng thái thiết lập kết nối chưa được tạo, kết nối được tạo khi một phương thức yêu cầu gởi dữ liệu đến hay nhận dữ liệu về từ server.
- Kết nối đi vào trạng thái đóng khi phương thức close() được gọi.
- Mặc dù không có phương thức nào của đối tượng Httpconnection có thể được dùng trong trạng thái đóng, nhưng các InputStream và OutputStream vẫn còn chứa dữ liệu.
- Các dòng InputStream và OutputStream vẫn có thể tiếp tục được dùng (sau khi thực hiện phương thức close) cho tới khi nó được đóng.

Câu 26: Kết nối đi vào trạng thái đóng khi phương thức close() được gọi thì các dòng dữ liệu InputStream và OutputStream có kết thúc không?

- Các InputStream và OutputStream vẫn còn chứa dữ liệu, các dòng này vẫn có thể tiếp tục được dùng cho tới khi chính nó được đóng.

Câu 27:Các yêu cầu về header trong kết nối HTTP (Phương thức, các thông tin về header tiêu biểu, header nào thực hiện yêu cầu userID-password?)

- Thiết lập (setup)

5.4. HTTP (tt)

Câu 26: Đáp ứng máy chủ (các thành phần và giải thích, các phương thức lấy và chuyển đổi đáp ứng).

- 3 thành phần là : dòng trạng thái chỉ ra kết quả của yêu cầu, header chỉ ra thuộc tính của đáp ứng dùng để thông dịch nội dung và body chứa nội dung của đáp ứng.

Câu 27: Các bước thực hiện kết nối HTTP với GET Request

1) Khởi tạo một đối tượng InputSteam.
2) Mở một kết nối HTTP với một URL từ xa.
3) Thiết lập các thuộc tính của kết nối.
4) Lấy đáp ứng HTTP_OK để biết kết nối đã được thiết lập hay chưa.
5) Nếu kết nối được thiết lập thì tiến hành lấy đối tượng InputStream từ kết nối để bắt đầu tải dữ liệu từ server.

Câu 28: Lớp nào được dùng để khởi tạo một đối tượng InputSteam để đọc dữ liệu đáp ứng?

- **Lớp OpenHttpConnection, cú pháp:**

InputStream OpenHttpConnection(String urlString)

5.4. HTTP (tt)

Cầu 29: Lớp nào được sử dụng để mở 1 kết nối http với URL từ xa và các phương thức nào được dùng để khai báo các thuộc tính cần thiết?

- Đối tượng **HttpURLConnection** để mở một kết nối HTTP với một URL từ xa. Ta thiết lập tất cả các thuộc tính khác nhau của kết nối, chẳng hạn như phương thức yêu cầu,..., như sau:

```
HttpURLConnection httpConn = (HttpURLConnection) conn;
httpConn.setAllowUserInteraction(false);
httpConn.setInstanceFollowRedirects(true); // tự động theo địa chỉ mới (true)
httpConn.setRequestMethod("GET");
```

- Có thể dùng lớp URLConnection và hàm openConnection(), cú pháp:

```
URL url = new URL(urlString);
URLConnection conn = url.openConnection();
```

Câu 30: Để biết kết kết đã được thiết lập hay chưa ta dùng hàm nào?

- Dùng hàm getResponseCode() và kiểm tra thông số HTTP_OK, cú pháp:

```
httpConn.connect();
response = httpConn.getResponseCode();
if (response == HttpURLConnection.HTTP_OK) {
in = httpConn.getInputStream();
}
```

5.4. HTTP (tt)

Câu 31: Để ứng dụng có thể kết nối Internet, ta cần bổ sung permission nào vào tập tin AndroidManifest.xml?

<uses-permission android:name="android.permission.INTERNET" />

Câu 32: Để tải dữ liệu nhị phân từ web vào ứng dụng, ví dụ: ta muốn tải một hình ảnh từ một máy chủ và hiển thị nó trong ứng dụng, ta cần thực hiện những bước nào?

- Thêm 1 ImageView vào layout của activity chính. Trong activity chính, ta khai báo một đối tượng ImageView, sau đó xây dựng thêm phương thức thực hiện chức năng tải ảnh về từ một trang web tương ứng với một URL:

```
private Bitmap DownloadImage(String URL){
Bitmap bitmap = null; InputStream in = null;
    try {
    in = OpenHttpConnection(URL); bitmap =
BitmapFactory.decodeStream(in); in.close();
    } catch (IOException e1) {
    Log.d("NetworkingActivity",
e1.getLocalizedMessage());
    }
    return bitmap;
    }
```

```
public class DownloadImageTask extends AsyncTask<String,
Void, Bitmap> {
    protected Bitmap doInBackground(String... urls) {
    return DownloadImage(urls[0]);
    }
    protected void onPostExecute(Bitmap result) {
    ImageView img = (ImageView) findViewById(R.id.img);
    img.setImageBitmap(result);
    }
    }
```

5.4. HTTP (tt)

Câu 32: Để tải dữ liệu văn bản từ web vào ứng dụng, ví dụ: ta muốn tải một hình ảnh từ một máy chủ và hiển thị nó trong ứng dụng, ta cần thực hiện những bước nào?

- Thêm 1 TextView vào layout của activity chính. Trong activity chính, ta khai báo một đối tượng TextView, sau đó xây dựng thêm phương thức thực hiện chức năng tải văn bản về từ một trang web tương ứng với một URL:

```
private String DownloadText(String URL){
int BUFFER_SIZE=2000; InputStream in = null;
    try {in = OpenHttpConnection(URL);}
   catch (IOException e) { Log.d("Networking",
e.getLocalizedMessage()); return ""; }
 InputStreamReader isr =new InputStreamReader(in);
 int charRead; String str = "";
 char[] inputBuffer = new char[BUFFER_SIZE];
 try {while ((charRead = isr.read(inputBuffer))>0) {
String readString =
          String.copyValueOf(inputBuffer, 0, charRead);
 str += readString;
inputBuffer = new char[BUFFER_SIZE];}
in.close();}
 catch (IOException e) {
  Log.d("Networking",e.getLocalizedMessage());
  return ""; }
  return str;   }
 private class DownloadTextTask extends
AsyncTask<String, Void, String> {
 protected String doInBackground(String... urls)
{return DownloadText(urls[0]); }
          //@Override
 protected void onPostExecute(String result) {
    Toast.makeText(getBaseContext(), result,
Toast.LENGTH_LONG).show() }
       }
```

5.4. HTTP (tt)

Câu 33: Các bước thực hiện giao thức HTTP với phương thức POST như sau:

1) Tạo ra một đối tượng HttpClient
2) Tạo ra một đối tượng HttpPost
3) Thêm vào các tham số của phương thức POST
4) Mã hoá dữ liệu POST
5) Thực hiện một HTTP POST request.

```
public class MainActivity extends ActionBarActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState); setContentView(R.layout.activity_main);
    String resultPost=""; HTTPPost httppost = new HTTPPost();
    List<NameValuePair> postParameters = new ArrayList<NameValuePair>();
    postParameters.add(new BasicNameValuePair("q", "cantho+university"));
    try {
    resultPost = httppost.executeHttpPost("www.ctu.edu.vn", postParameters);
    System.out.print(resultPost);  }
   catch (Exception e) {  e.printStackTrace(); }
  }
```

5.4. HTTP (tt)

```
public class HTTPPost {
   public String executeHttpPost(String uri, List<NameValuePair>  postParameters) throws
Exception {  BufferedReader in = null;   String result = "";
   try {
   HttpClient client = new DefaultHttpClient();  //Tạo ra một đối tượng HttpClient
  HttpPost request = new HttpPost(uri);  //Tạo ra đối tượng HttpPost.
  UrlEncodedFormEntity formEntity = new UrlEncodedFormEntity(postParameters);
  request.setEntity(formEntity);
  HttpResponse response = client.execute(request);
  in = new BufferedReader(new InputStreamReader(response.getEntity().getContent()));
 StringBuffer sb=new StringBuffer("");String line = "";
 String NL=System.getProperty("line.separator");
 while ((line = in.readLine()) != null) {  sb.append(line + NL);  }
 in.close(); result = sb.toString();
  } finally {  if (in != null) {  try { in.close(); } catch (IOException e) { e.printStackTrace(); }
  }  } return result;  } }}
```

5.5. JSON Service

Câu 34: JSON là gì?

- JSON (JavaScript Object Notation) là một kiểu dữ liệu mở trong JavaScript. Kiểu dữ liệu này bao gồm chủ yếu là text, có thể đọc được theo dạng cặp "thuộc tính - giá trị", là một định dạng trao đổi dữ liệu cỡ nhỏ dễ dàng cho việt đọc và viết. Nó là một kiểu dữ liệu trung gian, chủ yếu được dùng để vận chuyển thông tin giữa các thành phần của một chương trình.

Câu 35: Tại sao nên sử dụng JSON?

- Webservice thường lưu dữ liệu với MYSQL, SQL Server. Tuy nhiên trong các ứng dụng nhỏ, dữ liệu nhỏ gọn thì không cần dùng hệ quản trị dữ liệu lớn, phức tạp, khi đó ta có thể dùng JSON.

Câu 36: Cấu trúc ngôn ngữ JSON?

- Thông điệp JSON được chia thành các **dãy** (JSONArray) và **đối tượng** (JSONObject). Mỗi dãy được bắt đầu và kết thúc bởi cặp ngoặc vuông ([ ]) và mỗi đối tượng được bắt đầu và kết thúc bằng một cặp ngoặc quăn ({ }). Các **thông tin trong một đối tượng** bao gồm tập hợp các cặp khóa / giá trị (Key/Value).
- Ví dụ:

5.5. JSON Service

```
{
  “firstName”: “Hoa Minh”,
  “lastName”: “Doan”,
  “age”: 61,
  “gender”: “male”,
  “address”: {
    “streetAddress”: “Y31 Le Tan Quoc Street”,
    “Districk”: “Cai Rang”,
    “City”: “Can Tho”,
    “Country": "Viet Nam”
  },
  “phoneNumbers”: [
    {
      “type”: “Home”,
      “number”: "0710 3821897"
    },
    {
      “type”: “Mobile”,
      “number”: “0918628789”
    }
  ],
  "email": “dhminh@ctu.edu.vn”
}
```

Tập tin json có thể được lưu với bất kỳ phần mở rộng nào, tuy nhiên thông thường thì nó được lưu dưới phần mở rộng là .json hoặc .js.

5.5. JSON Service

Câu 36: Android hỗ trợ có lập trình dữ liệu với JSON không?

- Android hỗ trợ sẵn thư viện để làm việc với JSON (JSON API), ta không cần phải khai báo bất cứ một thư viện nào khác.

Câu 37: Lớp JSONObject có chức năng gì?

- Tạo ra đối tượng quản lý JSON ở dạng một Object.

Câu 38: Lớp JSONArray có chức năng gì?

- Tạo ra đối tượng quản lý JSON ở dạng tập hợn các Object hoặc Array.

Câu 39: Lớp JSONStringer có chức năng gì?

- Chuyển dữ liệu JSON thành dạng chuỗi.

Câu 40: JSONTokener có chức năng gì?

- Chuyển đổi đối tượng JSON (chuẩn RFC-4627) mã hoá chuỗi một thành đối tượng tương ứng.

5.5. JSON Service

`JSONObject()` Creates a `JSONObject` with no name/value mappings.
`JSONObject(Map copyFrom)` Creates a new `JSONObject` by copying all name/value mappings from the given map.
`JSONObject(JSONTokener readFrom)` Creates a new `JSONObject` with name/value mappings from the next object in the tokener.
`JSONObject(String json)` Creates a new `JSONObject` with name/value mappings from the JSON string.
`JSONObject(JSONObject copyFrom, String[] names)` Creates a new `JSONObject` by copying mappings for the listed names from the given object.

5.5. JSON Service

JSONArray	getJSONArray(int index) Returns the value at index if it exists and is a JSONArray.
JSONObject	getJSONObject(int index) Returns the value at index if it exists and is a JSONObject.
JSONObject	accumulate(String name, Object value) Appends value to the array already mapped to name.
JSONArray	getJSONArray(String name) Returns the value mapped by name if it exists and is a JSONArray, or throws otherwise.
JSONObject	getJSONObject(String name) Returns the value mapped by name if it exists and is a JSONObject, or throws otherwise.
JSONArray	names() Returns an array containing the string names in this object.

5.5. JSON Service

`JSONArray()`

Creates a `JSONArray` with no values.

`JSONArray(Collection copyFrom)`

Creates a new `JSONArray` by copying all values from the given collection.

`JSONArray(JSONTokener readFrom)`

Creates a new `JSONArray` with values from the next array in the tokener.

`JSONArray(String json)`

Creates a new `JSONArray` with values from the JSON string.

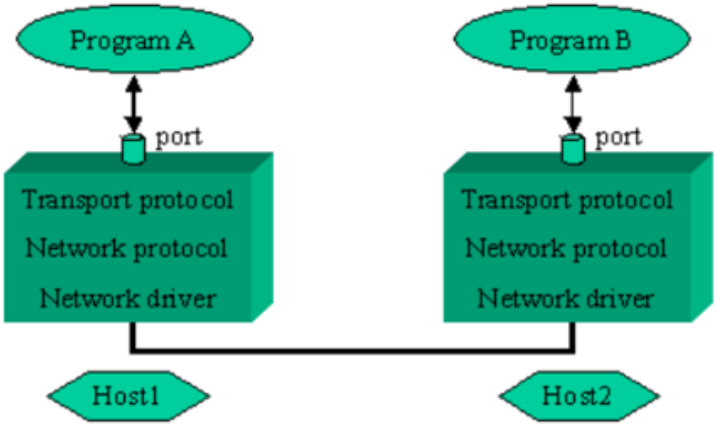
`JSONArray(Object array)`

Creates a new `JSONArray` with values from the given primitive array.

5.6. SOCKET

Câu 41: Kết nối socket là loại kết như thế nào?

- Socket là loại kết nội mạng cho phép thiết lập các kênh giao tiếp mà hai đầu kênh được đánh dấu bởi hai cổng (port). Thông qua các cổng này một quá trình có thể nhận và gởi dữ liệu với các quá trình khác.



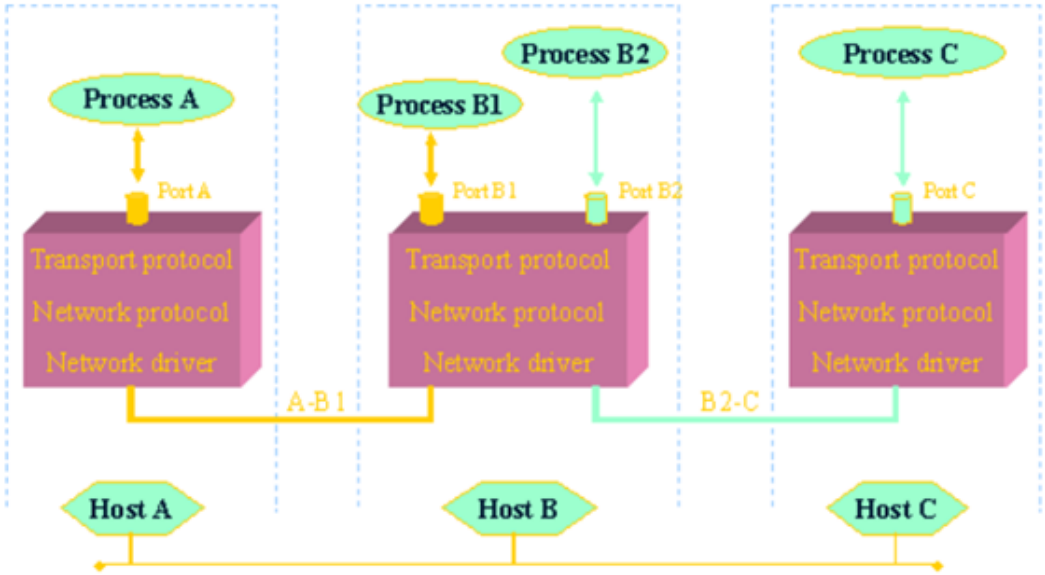
Câu 42: Có các kiểu kết nồi socket nào?

- Có hai kiểu socket:
- 1. Socket kiểu AF_UNIX chỉ cho phép giao tiếp giữa các quá trình trong cùng một máy tính.
- 2. Socket kiểu AF_INET cho phép giao tiếp giữa các quá trình trên những máy tính khác nhau trên mạng.

5.6. SOCKET

Câu 43: Để có thể thực hiện các cuộc giao tiếp, một trong hai quá trình phải công bố số hiệu cổng của socket mà mình sử dụng. Ngoài số hiệu cổng, hai bên giao tiếp còn phải biết địa chỉ IP của nhau. Cjo biết vai trò của số hiệu cổng và địa chỉ IP.

- Mỗi cổng giao tiếp thể hiện một địa chỉ xác định trong hệ thống. Khi quá trình được gán một số hiệu cổng, nó có thể nhận dữ liệu gởi đến cổng này từ các quá trình khác. Địa chỉ IP giúp phân biệt các thiết bị trên mạng TCP/IP. Trong khi số hiệu cổng dùng để phân biệt các quá trình khác nhau trên cùng một máy tính.



Trong hình bên, địa chỉ của quá trình B1 được xác định bằng 2 thông tin (Host B và port B1): Địa chỉ máy tính có thể là địa chỉ IP dạng 203.162.36.149 hay là địa chỉ tên miền như www.cit.ctu.edu.vn

5.6. SOCKET

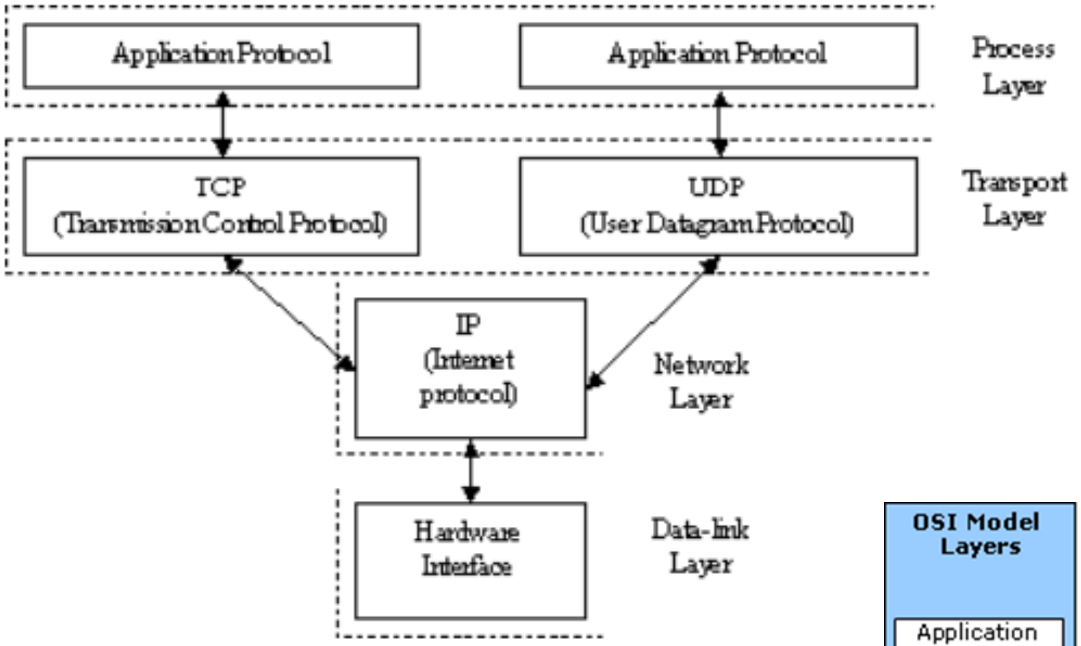
Câu 44: Cho biết các qui định về số hiệu cổng?

- Số hiệu cổng gán cho Socket phải duy nhất trên phạm vi máy tính/thiết bị di động đó trong khoảng từ 0 đến 65535 (16 bits).
- Trong đó, các giá trị cổng được qui định như sau:
  - Từ 0 đến 1023: được gọi là cổng hệ thống (common hay well-know ports) được dành riêng cho các quá trình của hệ thống.
  - Tử 1024 đến 49151: được gọi là cổng phải đăng ký (registered port). Về mặt nguyên tắc, các ứng dụng sử dụng các cổng này phải đăng ký với IANA (Internet Assigned Numbers Authority).
  - Từ 49152 đến 65535: được gọi là cổng dùng riêng hay cổng động (dynamic-private port), ta có thể dùng cho các ứng dụng mà không cần đăng ký hoặc hệ thống dùng để gán tự động cho quá trình client.

Câu 45: Ch biết các cổng mặc định của 1 số dịch vụ mạng thông dụng?

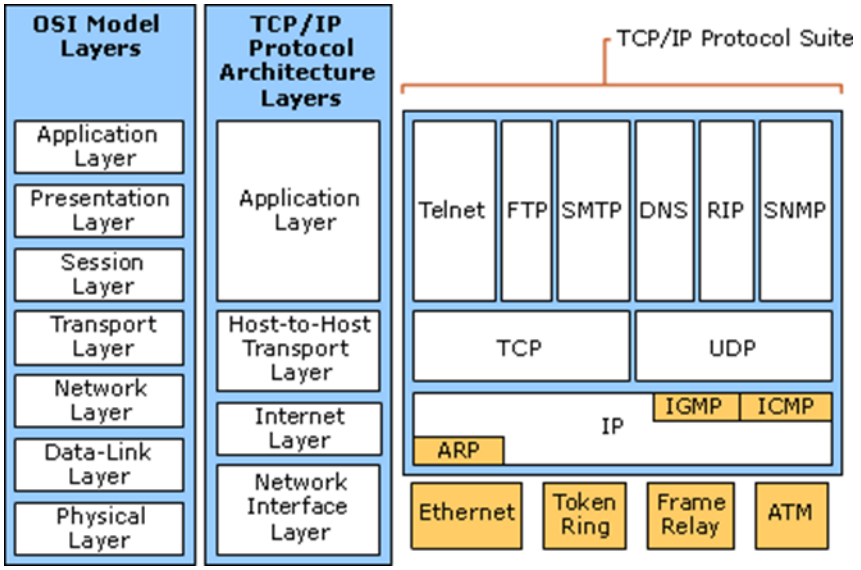
Số hiệu cổng	Quá trình hệ thống	Số hiệu cổng	Quá trình hệ thống
7	Dịch vụ Echo	25	Dịch vụ email (SMTP)
21	Dịch vụ FTP	80	Dịch vụ WEB (HTTP)
23	Dịch vụ Telnet	110	Dịch vụ email (POP)

5.6. SOCKET



Bộ giao thức TCP/IP

Mô hình 4 tầng của TCP / IP và 7 tầng của mô hình OSI (Một giao thức liên mạng đã cũ và bị thay thế bởi TCP/IP)



5.6. SOCKET

Câu 47: Phân biệt các kiểu kết nối mạng HTTP và Socket?

- **Socket** là giao diện giữa chương trình ứng dụng với tầng vận chuyển. Nó cho phép ta chọn giao thức sử dụng ở tầng vận chuyển là TCP hay UDP cho chương trình ứng dụng của mình.
- **HTTP** là giao thức nằm trong tầng **Application Layer**, được sử dụng để truyền nội dung trang Web từ Web Server đến trình duyệt Web ở Client. Là giao thức Client/Server dùng cho Internet - World Wide Web, HTTP thuộc tầng ứng dụng của bộ giao thức TCP/IP (Các giao thức nền tảng cho Internet).

Câu 48: HTTP/Socket dùng trong các trường hợp nào?

- Dịch vụ web sử dụng giao thức HTTP để truyền thông tồn tại một bất lợi là thiếu tính "ổn định", khi kết nối với một dịch vụ web bằng giao thức HTTP, mỗi kết nối được thiết lập đều được xem như là một kết nối mới - web server không duy trì một kết nối liên tục với client.
- Nếu ta muốn xây dựng một ứng dụng mà nó có thể duy trì kết nối liên tục đến server và những thay đổi ở phía server đều được thông báo đến client. Khi đó, ta cần phải sử dụng socket.

5.6. SOCKET

Câu 49: Phân biệt TCP và UDP?

Chế độ có kết nối (TCP)	Chế độ không kết nối (UDP)
▪ Tồn tại (thiết lập và duy trì) kênh giao tiếp giữa 2 bên giao tiếp. ▪ Dữ liệu được gởi đi theo chế độ bảo đảm: có kiểm tra lỗi, truyền lại gói tin lỗi hay mất, bảo đảm thứ tự đến của các gói tin,… dữ liệu nhận được chính xác. ▪ Chiếm nhiều tài nguyên mạng, tốc độ truyền chậm. ▪ Thích hợp cho các ứng dụng như: truyền tập tin, trực tuyến,… những dịch vụ mà chiều dài dữ liệu không được định trước.	▪ Không có sự kết nối trước giữa 2 bên giao tiếp (giả sử đầu nhận luôn sẵn sàng). ▪ Dữ liệu gởi đi theo chế độ không bảo đảm: không phát hiện và không truyền lại gói tin lỗi hay bị mất, không bảo đảm thứ tự đến của các gói tin,… dữ liệu nhận được không chính xác. ▪ Không chiếm nhiều tài nguyên của hệ thống, tốc độ truyền nhanh. ▪ Thích hợp cho các ứng dụng như: truyền âm thanh, hình ảnh, thông báo giờ, tỉ giá, tin nhắn,…

5.6. SOCKET

Câu 50: Trình bày các bước để thực hiện một kết nối socket giữ client và server với giao thức TCP.

- Bước 1: Server tạo socket, gán số cổng và lắng nghe yêu cầu kết nối từ client, gồm các phương thức: socket(), bind(), listen()
- Bước 2: Client tạo socket yêu cầu thiết lập một kết nối đến server, gồm các phương thức: socket, connect(), accept()
- Bước 3: đóng kết nối

Câu 52: Trình bày các bước để thực hiện một kết nối socket giữ client và server với giao thức UDP.

- Bước 1: Server tạo socket, gán số cổng, gồm các phương thức: socket(), bind(),
- Bước 2: Client tạo socket yêu cầu thiết lập một kết nối đến server, gồm các phương thức: socket, connect(), accept()
- Bước 3: đóng kết nối

5.6. SOCKET

Câu 53: Trong lập trình ứng dụng Android, để thực hiện một kết nối socket giữa Client – Server ta cần sử dụng các lớp nào và tiến hành những bước nào?

- Các lớp trong các thư viện cần import:

- import java.io.BufferedReader; → để khai báo 1 bộ đệm đọc message.
- import java.io.InputStreamReader; → để tạo 1 luồng đọc dữ liệu.
- import java.io.OutputStream; → để tạo 1 luồng gởi dữ liệu.
- import java.net.ServerSocket; → để mở Server Socket
- import java.net.Socket; → để mở Client Socket
- import android.os.Handler; → Tạo ra đối tượng Hander để điều hiển việc gởi và đọc message.
- import android.os.Message; → để khai báo đối tượng message.

- Các bước tiến hành:

- Khai báo 1 Server Socket → Khai báo Client Socket → Khai báo các luồng ra/vào ra → Khai báo 1 bộ đệm để đọc message → tạo 1 tiến trình (Threat) để thực hiện kết nối → bổ sung thêm permission vào tập tin AndroidManifest.xml

5.7. Google Maps và GPS

Câu 54: Google Maps là một trong các ứng dụng đi kèm với nền tảng Android. Ngoài việc sử dụng ứng dụng Maps có sẵn trên smartphone, trong nhiều trường hợp ta cần nhúng nó vào ứng dụng của. Để lập trình khai thác Google Maps trong ứng dụng, ta sử dụng View nào và đượ khai báo trong tập tin layout như thế nào?

- **MapView** được dùng để hiển thị bản đồ, có thể khai báo trong tập tin layout XML hoặc trong mã của chương trình Java.
- MapView được khai báo trong tập tin Layout XML như sau:

```
<com.google.android.maps.MapView
    android:id="@+id/map"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:clickable="true"
    android:apiKey="API key đã xác định"/>
```

5.7. Google Maps và GPS

Câu 55: Để lập trình khai thác Google Maps ta cần khai báo các thư viện nào?

- **import** com.google.android.maps.GeoPoint;
- **import** com.google.android.maps.MapActivity;
- **import** com.google.android.maps.MapController;
- **import** com.google.android.maps.MapView;

Câu 56: Google Maps hỗ trợ các kiểu xem bản đồ nào? Câu lệnh?

- Các kiểu: vệ tinh (Satellite), đường phố (StreetView) và giao thông (Traffic), tương ứng với các câu lệnh sau:

- *map.setSatellite(false);*
- *map.setStreetView(false);*
- *map.setTraffic(false);*

- Để xem dạng bản đồ nào chỉ cần đặt đối số cho nó là true.

Câu 57: Câu lệnh có tác dụng phóng to/thu nhỏ bản đồ?

- phóng to bản đồ: *map.getController().zoomIn();*
- thu nhỏ bản đồ: *map.getController().zoomOut();*

5.7. Google Maps và GPS

Câu 58: Để di chuyển đến 1 toạ độ ta phải khai báo đối tượng gì và dùng phương thức nào?

- phải khai báo đối tượng MapController và dùng hàm animateTo().

Câu 59: muốn Google Maps hiển thị ta cần xác định được APIKey của nó, cách làm như thế nào?

- Mở **cmd** và dùng lệnh **cd** di chuyển tới thư **mục bin** của **JDK** (ví dụ C:\Program Files\Java\jdk1.6.0_22\bin). → gõ vào lệnh sau ***keytool.exe -v -list -alias androiddebugkey -keystore "đường dẫn\debug.keystore" -storepass android -keypass android.*** Trong đó *đường dẫn\debug.keystore* là đường dẫn tới tập tin debug.keystore trong máy tính mặc định thường là **C:\Users\tên tài khoản\.android.** → Khi nhận được mã MD5 của keystore truy cập vào trang web ***http://code.google.com/android/maps-api-signup.html***. Nhập mã MD5 vào ô trống, đánh dấu chọn mục I have read... và nhấn Generate API key để nhận API key. Lưu key này lại và kể từ đây về sau mỗi khi cần tạo một control MapView thì chỉ cần mở ra sử dụng nó.

5.7. Google Maps và GPS

Câu 60: Giới thiệu về GPS?

Hệ thống định vị toàn cầu (Global Positioning System - GPS) là hệ thống xác định vị trí dựa trên vị trí của các vệ tinh nhân tạo, do Bộ Quốc phòng Hoa Kỳ thiết kế, xây dựng, vận hành và quản lý. Trong cùng một thời điểm, tọa độ của một điểm trên mặt đất sẽ được xác định nếu xác định được khoảng cách từ điểm đó đến ít nhất ba vệ tinh. Tuy được quản lý bởi Bộ Quốc phòng Hoa Kỳ, nhưng chính phủ Hoa Kỳ cho phép mọi người trên thế giới sử dụng một số chức năng của GPS miễn phí, bất kể quốc tịch nào.

Câu 61: : Để sử dụng GPS trên Android, trong tập tin AndroidManifest.xml ta bổ sung thêm permission gì?

```
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"></uses-permission>
```

Câu 62: Để GPS cập nhật một vị trí mới, ta dùng đối tượng nào và ghi đè vào phương thức public void onCreate(Bundle savedInstanceState) như thế nào?

- Ta cần bổ sung một LocationListener, như sau:

```
LocationManager mlocManager =
(LocationManager)getSystemService(Context.LOCATION_SERVICE);
LocationListener mlocListener = new MyLocationListener();
mlocManager.requestLocationUpdates( LocationManager.GPS_PROVIDER, 0, 0,mlocListener);
```

5.7. Google Maps và GPS

Câu 63: Ngoài ra còn có những phương thức nào có thể khai thác khi lập trình GPS?

- onLocationChanged (Cập nhật vị trí);
- onProviderDisabled (Làm khi GPS được vô hiệu hóa);
- onProviderEnabled (Làm khi GPS được kích hoạt);
- onStatusChanged (Làm khi thay đổi trạng thái, thường thì ít được sử dụng).