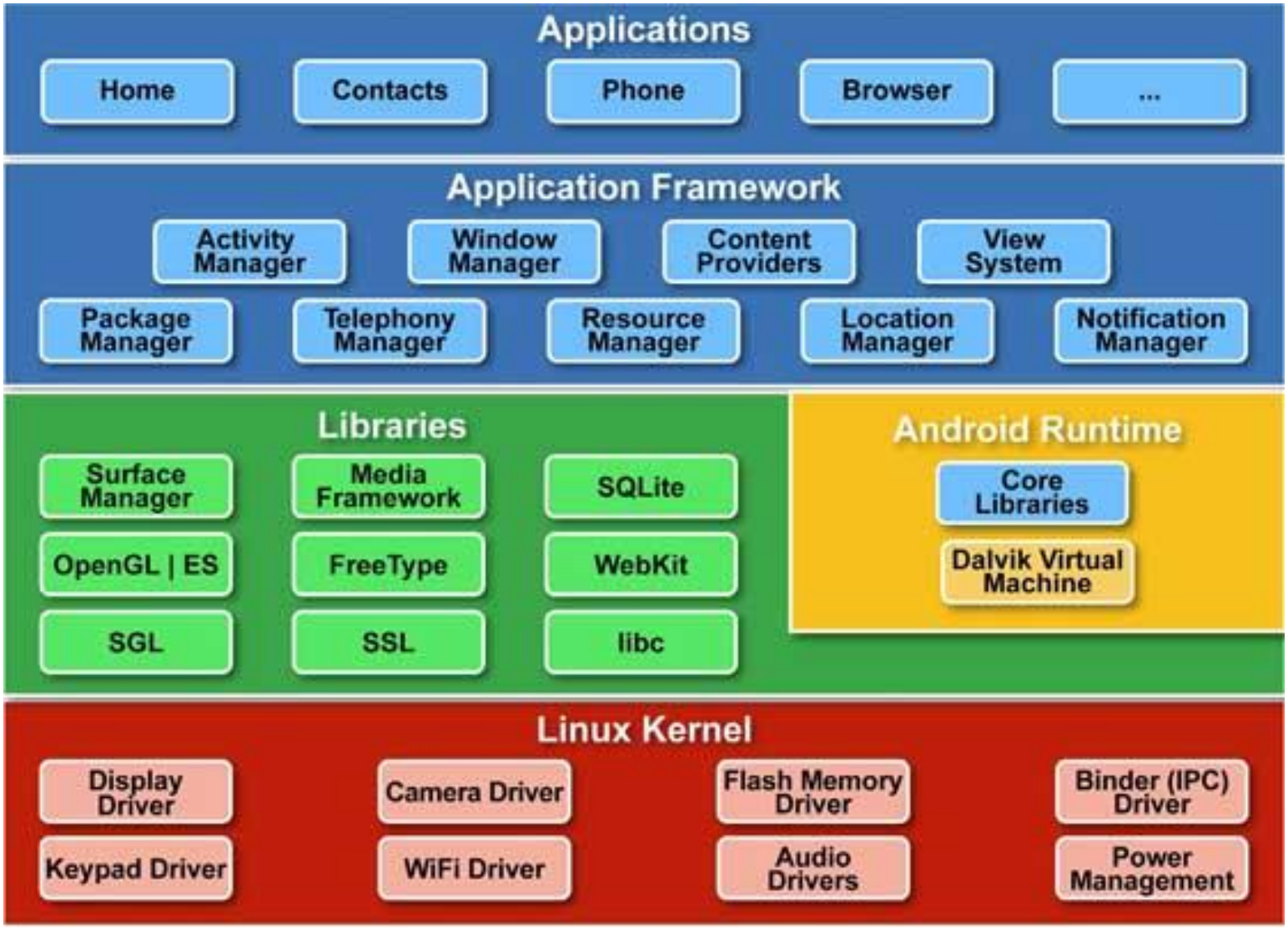
ANDROID PROGRAMMING LESSON 2

Version 1.0

Agenda

- Android Architecture
- Application Components
- Activity Lifecycles
- Handling Android Activity State Changes
- Saving and Restoring the State of an Android Activity

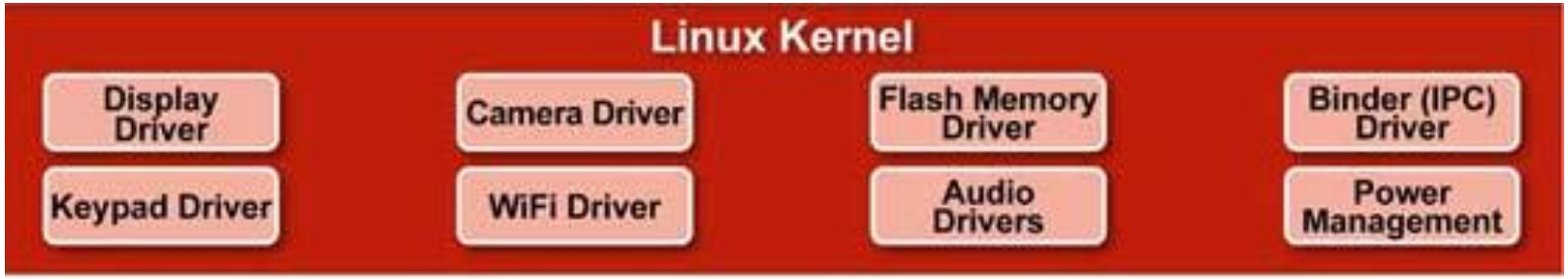
Android Architecture



Android Architecture

- **Linux kernel**

  provides a low level of abstraction between the device hardware and the upper layers of the Android software stack
- Contains all low level drivers for various hardware components support.



Android Architecture

- **Android Runtime (ART)**

  Designed to run apps in a constrained environment that has limited muscle power in terms of battery, processing and memory.

Android Architecture

- **Libraries**
  - Exposed to developers through Android Application Framework.
  - Contains C/C++ libraries used by components of Android Systems.

Android Architecture

- **Application Framework**

  It is a collection of APIs written in Java, which gives developers access to the complete feature set of Android OS.

Android Architecture

- **Applications**

  Top of the Android Application Stack, is occupied by the System apps and tonnes of other Apps that users can download from Android's Official Play Store, also known as Google Play Store.

Application Components

- **Android Activities**
  - A subclass of Activity class
  - A single user interface screen
  - Reuseable

Application Components

- **Android Intents**

  Intents are the mechanism by which one activity is able to launch another and implement the flow through the activities that make up an application

  - *Explicit intent*
  - *Implicit intent*

Application Components

- **Broadcast Intents**

  Is a system wide intent that is sent out to all applications that have registered an “interested” *Broadcast Receiver*

```
<activity android:name=".ExampleActivity"
android:icon="@drawable/app_icon">
    <intent-filter>
        <action android:name="android.intent.action.SEND"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:mimeType="text/plain"/>
    </intent-filter>
</activity>
```

Application Components

- **Android Services**
  - Android Services are processes that run in the background and do not have a user interface
  - Android Services are ideal for situations where an application needs to continue performing tasks but does not necessarily need a user interface to be visible to the user

Application Components

- **Content Providers**
  - Content Providers implement a mechanism for the sharing of data between applications
  - Data can be shared in the form of a file or an entire SQLite database

Application Components

- **Application Manifest**

  The glue that pulls together the various elements that comprise an application is the Application Manifest file

Application Components

- **Application Resources**
  - Android application package will also typically contain a collection of *resource files*.
  - These files contain resources such as the strings, images, fonts and colors that appear in the user interface together with the XML representation of the user interface layouts

Application Components

- **Application Context**
  When an application is compiled, a class named *R* is created that contains references to the application resources. The application manifest file and these resources combine to create what is known as the *Application Context*. This context used in the application code to gain access to the application resources at runtime

Activity Lifecycles

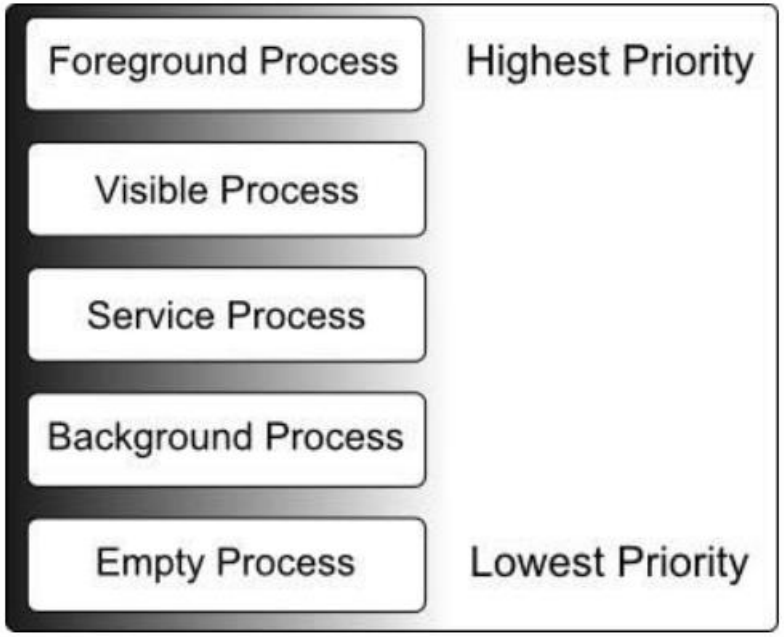
- **Android Applications and Resource Management**

  If the system identifies that resources on the device are reaching capacity it will take steps to terminate processes to free up memory. the system base on *priority* and *state* to determine which process to terminate.

Activity Lifecycles

- **Android Process States**

  The current state of a process is defined by the highest-ranking active component



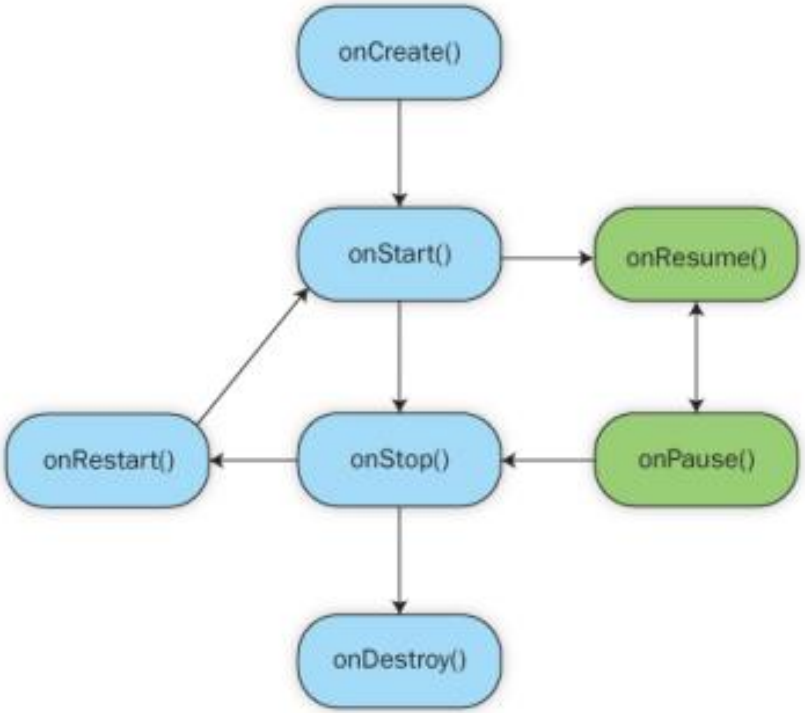
Activity Lifecycles

- **Inter-Process Dependencies**
  states of a process can never be ranked lower than another process that it is currently serving. *Ex a service process acting as the content provider for a foreground process*

Activity Lifecycles

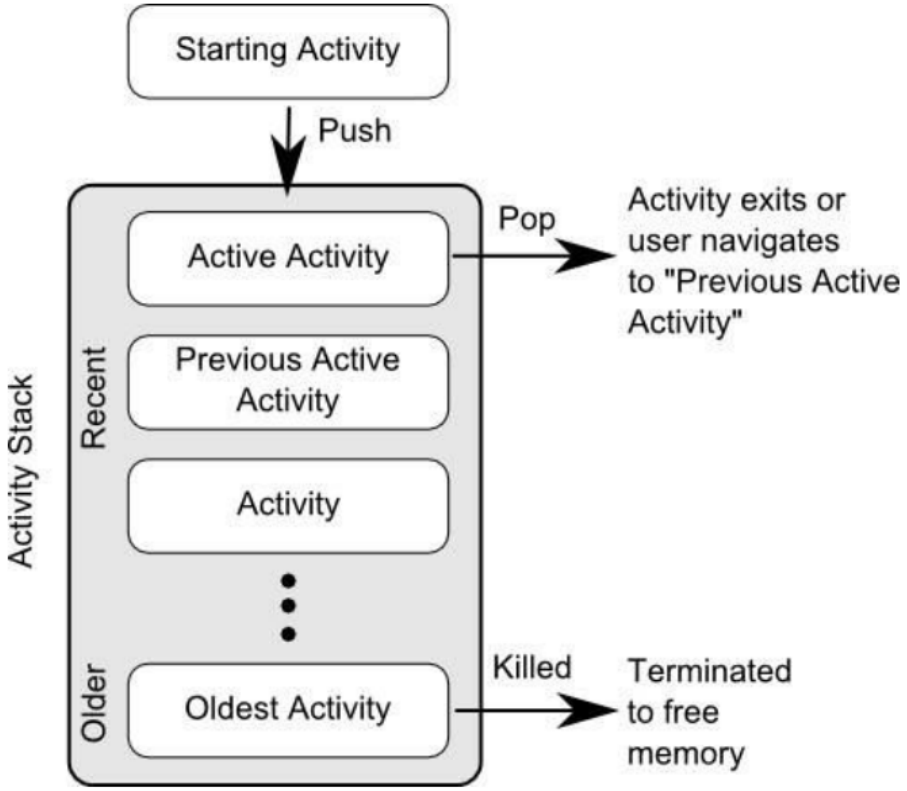
- **The Activity Lifecycle**

  activities also transition through different states during the execution lifetime of an application.



Activity Lifecycles

- **The Activity Stack**



Activity Lifecycles

- **Activity States**
  - Active / Running
  - Paused
  - Stopped
  - Killed

Activity Lifecycles

- **Configuration Changes**

  Configuration change that impacts the appearance of an activity

- **Handling State Change**

  Application will, in most circumstances, be notified by the runtime system of the changes and given the opportunity to react accordingly

Handling Activity State Changes

- **The Android Activity Lifecycle Methods**

  The Activity class contains a number of lifecycle methods which act as event handlers when the state of an Activity changes. Each change has a support method.

Handling Activity State Changes

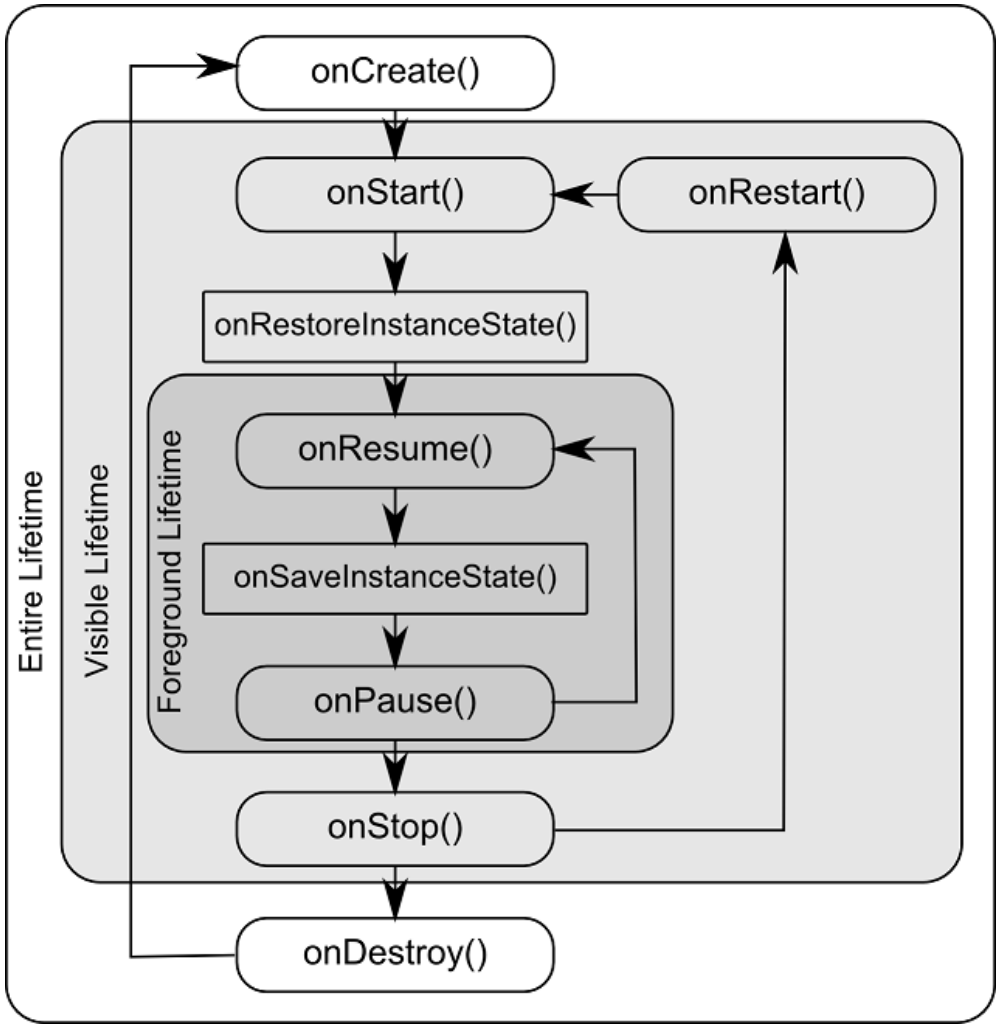
- **Activity Lifetimes**

  Entire Lifetime –The term “entire lifetime” is used to describe everything that takes place within an activity between the initial call to the *onCreate()* method and the call to *onDestroy()* prior to the activity terminating.

  Visible Lifetime – Covers the periods of execution of an activity between the call to *onStart()* and *onStop()*. During this period the activity is visible to the user though may not be the activity with which the user is currently interacting.

  Foreground Lifetime – Refers to the periods of execution between calls to the *onResume()* and *onPause()* methods

Handling Activity State Changes



Disabling Configuration Change Restarts

- Activity may indicate that it is not to be restarted in the event of certain configuration changes. This is achieved by adding an *android:configChanges* directive to the manifest file of the activity

```
<activity android:name=".DemoActivity"
          android:configChanges="orientation|fontScale"
          android:label="@string/app_name">
```

Saving and Restoring the State of an Android Activity

- **The Bundle Class**
  The Bundle class also contains a set of methods that can be used to get and set key-value pairs for a variety of data types

Saving and Restoring the State of an Android Activity

- **Saving the State**

```
@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    final EditText textBox = (EditText) findViewById(R.id.editText);
    CharSequence userText = textBox.getText();
    outState.putCharSequence("savedText", userText);
}
```

- **Restoring the State**

```
@Override
protected void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);

    final EditText textBox = (EditText) findViewById(R.id.editText);
    CharSequence userText = savedInstanceState.getCharSequence("savedText");
    textBox.setText(userText);
}
```

Saving and Restoring the State of an Android Activity

- Automatic saving and restoring

```
<EditText
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:saveEnabled="true"
    android:text="Name"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Quiz

- ***Android: which attribute will be used in XML if the Java code needs a reference to View?***
  A. android: gravity
  B. android: background
  C. android: index
  D. android: id
- ***The application must declare all its component in Manifest file?***
  A. True
  B. False

- ***The px unit corresponds to an actual pixcel on screen?***

  A. True

  B. False

- ***Which method is NOT a method of activity class?***

  A. onCreate()

  B. onStart()

  C. onPlay()

  D. onResume()

  E. onDestroy()

- ***How do you specify the minimum version of Android required by your application?***

  A. You specify the minimum Android version required using the minNdkVersion attribute in the AndroidManifest.xml file

  B. You specify the minimum Android version required using the minNumberVersion attribute in the AndroidManifest.xml file 

  C. You specify the minimum Android version required using the minSdkVersion attribute in the AndroidManifest.xml file

  D. You specify the minimum Android version required using the minVersion attribute in the AndroidManifest.xml file

- what layout is good for stacking views vertically or horizontally?
- this layout allows positions of view relative to other views?
- android apps run by default on the ___ thread?
- the___ thread handles all of the user input and output?
- What kind of thread should handle long running processes?

- End of Lesson 2

Thank you!