Devarshi Patel
UBID:50060983

CSE 486/586 Distributed Systems (Programming Assignment 1)

Simple Messenger on Android

A simple messenger application for android platform is implemented using two android virtual devices(AVD's), viz. avd0 and avd1. Initially the Android Development Toolkit was downloaded and extracted, in order to create an android application, using the platform tools and other necessary provided in this ADT bundle. Path to the tools and platform tools was appended to system path, so as to have access to tools, when required. A Python script, create_avd.py was used in order to create these AVDs and run_avd.py file was executed so as to run the AVDs, and the script set_redir 10000.py was set in order to establish connection between the two AVDs being used.

Simple Messenger is and android application which consists of MainActivity class, and two threads in this class. The two threads are client thread and server thread. Both these classes extends AsyncTask. AsyncTask is used in order to perform background tasks, and this subclass will override doInBackground(params...) method. AsyncTask has 4 methods, viz onPreExecute(), doInBackground(), onProgressUpdate(), and onPostExecute(). If one wants to perform some previous tasks before executing doInBackground(), those tasks should be written in onPreExecute() method. Call to publishProgress(Progress...) invokes onProgressUpdate(Progress...), which can be called in order to update the current progress.

Main Activity:
In this application, we define a EditText, where message to be sent can be entered and a TextView, to display the received message from the sender, in the onCreate() method of the MainActivity class. The Server thread is instantiated, also a key listener is set for the EditText, using setOnKeyListener, in order to perform key events after typing a message in the EditText field. We use keycode to identify whether the key event was enter or not, if yes, then the client thread is called using the executeOnExecutor() method, which provides parallel execution. TelephonyManager is also used in order to get Telephony service, using the object of TelephonyManager, we can obtain the telephone number, the substring of which can be used in order to identify the AVDs, and which in turn can help in socket creation, that is if the substring is “5554”, it signifies that AVD0 is used and thus a socket is created by passing the IP address and port number of AVD1 (i.e. 11112), thus AVD0 will act as a client and AVD 1 as server, or vice versa.

Client Thread:
The client thread is used to send message to the server thread and this can be done if and only if the connection between client and server can be made successfully. The client thread extends the AsyncTask, which makes handling of UI threads easier. Here, the processing is done in the doInBackground() method provided by the AsyncTask, and socket creation takes place based on the present active AVD, which would act as client making the other AVD server, which would listen to requests. DataOutputStream is used in order to perform write operation to server, so the message which was extracted from the EditText using getText() function and converting it to string using toString() function, is written to the network using the output object of DataOutputStream class and also the

getOutputstream() method to send message into the network and the writeBytes() function of this class can be used in order to perform write on the server.

Server Thread:
The server thread is used to listen and accept client connections and also to retrieve the requests made by the clients. The server thread also extends the AsyncTask and the listening and acceptance of connection and all the processing is done in the doInBackground() method of this AsyncTask. Also the InputStreamReader is used to retrieve data or message sent by the client, the getInputStream() is used to get input stream and using the object of class InputStreamReader, we can read message using readLine() function of this class. Now, after getting the message it needs to be displayed on the TextView, thus the Text View needs to be updated and this can be done very effectively using the publishProgress() method of the AsyncTask. The publishProgress() takes message received from the client as a parameter and makes a call to onProgressUpdate() which can update the TextView of the server and set its TextView to the message received. Thus the message is successfully received at the other end.

Logs are also used in this code in order to print and understand the working of this code and logs make debugging easier, and thus prove helpful. Log.v provides the detailed log.

The user enters message, and after typing in the message the user presses enter key, this key event is detected and which would result in client thread initiation which would then identify the AVD number trying to act as client by comparing the substring of the telephone number received by using the object of TelephonyManager. It sets the connection between two AVDs. The set_redir.py 10000 is used in order to set the port numbers of AVD0 to 11108 and AVD1 to 11112 implicitly and thus message when delivered would map to the corresponding port and thus message would be delivered to other end.

Creating a simple messenger helped me to understand the basics of android application development, and I learned the installation and environment setup for ADT, usage of MainActivity class, functions of MainActivity, and also gained some knowledge about AsyncTask.

Communication over Network: