Md Rahman

1. What is interprocess communication?

2. Define each of the following and a short description of how each is implemented:

AIDL

Messenger (for IPC)

Binder

3. What are the restrictions for background services and what are some ways to accommodate for the restrictions?

4. What are the restrictions for broadcast receivers?

5. What is client/server relationships?

1. What is interprocess communication?

IPC refers to a mechanism where the OS allows various process to communicate with one another, synchronizing their actions and managing shared data. A process may let another process know that an event has occurred, or data may be transferred between them.

2. Define each of the following and a short description of how each is implemented:

1. AIDL. Android Interface Definition Language. Allows for defining of the programming interface agreed upon by both client and server in order to communicate via IPC. A process can’t typically access another’s memory, so they decompose their objects into primitives that the OS can recognize, and then marshall (transform the memory representation of an object to a data format that can be stored or transferred) that object. The AIDL handles this marshalling.

We define our AIDL interface in an .aidl file, save it in the source code directory of both the application hosting the service and any other application that binds to the service. Afterwards, the Android SDK generates an IBinder interface, and the client apps can bind to the service and call methods from the IBinder to perform IPC.

1. Messenger. A messenger is a reference to a handler. A handler allows for the sending and processing of messages and runnable objects. Messengers allows for message-based communication across processes. To implement a messenger, we need a service handler to handle incoming requests ,and the messenger itself, which implements the IBinder interface so a client can interact with the service.
2. Binder. An Android specific interface, where one android process may call a routine in another process using Binder to identify the necessary method to call, then pass the arguments between the processes. We use the tansact() API, which provides methods for sending a call to an IBinder object and receiving a call coming in to a Binder object. This API is synchronous.

3. What are the restrictions for background services and what are some ways to accommodate for the restrictions?

When multiple apps are running at once, the system takes a hit on both its performance and battery. This is also the case for when an app is running in the background (like music). When an app is in the background, it has a limit for how long it can continue running. At the end of that time, it’s considered idle, and it will either notify the user or the system will completely stop it. Broadcast receivers must be explicitly stated in the code, no longer implicitly defined. This is so the broadcast receivers aren’t constantly listening for signals, and battery and resources can be saved.

To get around such things, we can use job schedulers which allows an application to work while not being on an active run.

4. What are the restrictions for a broadcast receiver?

Apps must register to receive specific broadcasts, either thru declaring them in the manifest or registering with a context.

A Local broadcast Manager restricts broadcast to its own application.

System Receivers is when a system application broadcasts internally to all apps that are registered to it (such as airplane mode).

5. What is a client/server relationship?

The means of communication between a client application and the service application. The server application provides the information and the client receives it. We define how information is sent and how threads communicate with each other.