Ex. No. 7	**Running Multiple Operations Using Foreground Services**
Date of Exercise	04 - 10 - 2024

Aim

The aim of this experiment is to develop an Android app that uses a foreground service to concurrently play a video and download a file.

Program

MainActivity.kt

```
package com.example.exp_7
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.content.IntentFilter
import android.net.Uri
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.localbroadcastmanager.content.LocalBroadcastManager
import com.example.exp_7.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        binding.startServiceButton.setOnClickListener {
            val intent = Intent(this, ForegroundService::class.java)
            startService(intent)
        }
        LocalBroadcastManager.getInstance(this)
            .registerReceiver(videoReceiver, IntentFilter("PlayVideo"))
    }

    private val videoReceiver: BroadcastReceiver = object : BroadcastReceiver() {
        override fun onReceive(context: Context?, intent: Intent?) {
            val videoUrl = intent?.getStringExtra("video_url")
            videoUrl?.let {
                binding.videoView.setVideoURI(Uri.parse(it))
                binding.videoView.start()
            }
        }
    }
    override fun onDestroy() {
        super.onDestroy()
        LocalBroadcastManager.getInstance(this).unregisterReceiver(videoReceiver)
    }
}
```

activity_main.xml

```
<!-- res/layout/activity_main.xml -->
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <VideoView
```

```
        android:id="@+id/videoView"
        android:layout_width="wrap_content"
        android:layout_height="263dp" />

    <Button
        android:id="@+id/startServiceButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start Foreground Service"
        android:layout_gravity="center"/>
</LinearLayout>
```

ForegroundService.kt

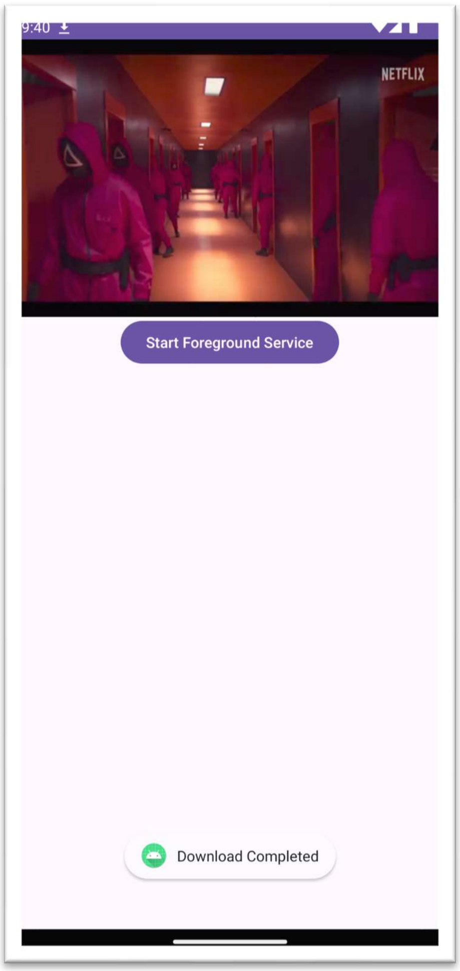
```
package com.example.exp_7
import android.app.*
import android.content.*
import android.media.MediaPlayer
import android.net.Uri
import android.os.Build
import android.os.Environment
import android.os.IBinder
import android.widget.Toast
import androidx.core.app.NotificationCompat
import androidx.localbroadcastmanager.content.LocalBroadcastManager

class ForegroundService : Service() {
    private val CHANNEL_ID = "ForegroundServiceChannel"
    private lateinit var mediaPlayer: MediaPlayer
    private lateinit var downloadManager: DownloadManager
    private var downloadId: Long = 0
    private val FILE_URL =
"https://drive.google.com/uc?export=download&id=1SD55mNItmrbyeADhojS1EjMlrCtruof
Y"
    override fun onCreate() {
        super.onCreate()
        createNotificationChannel()

        mediaPlayer = MediaPlayer()
        mediaPlayer.setOnCompletionListener {
            stopSelf()
        }
    }
    override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
        downloadFile()

        val videoIntent = Intent("PlayVideo")
        videoIntent.putExtra("video_url",
"android.resource://${packageName}/raw/videoplayback")
        LocalBroadcastManager.getInstance(this).sendBroadcast(videoIntent)

        val notificationIntent = Intent(this, MainActivity::class.java)
        val pendingIntent = PendingIntent.getActivity(this, 0, notificationIntent,
PendingIntent.FLAG_IMMUTABLE)

        val notification = NotificationCompat.Builder(this, CHANNEL_ID)
            .setContentTitle("Foreground Service")
            .setContentText("Downloading file")
            .setSmallIcon(R.drawable.img)
            .setContentIntent(pendingIntent)
            .build()

        startForeground(1, notification)
        return START_NOT_STICKY
    }

    private fun downloadFile() {
        val uri = Uri.parse(FILE_URL)
        val request = DownloadManager.Request(uri)
        request.setDestinationInExternalPublicDir(Environment.DIRECTORY_DOWNLOADS,
"largefile.zip")

request.setNotificationVisibility(DownloadManager.Request.VISIBILITY_VISIBLE_NOTIFY
_COMPLETED)

        downloadManager = getSystemService(Context.DOWNLOAD_SERVICE) as
DownloadManager
        downloadId = downloadManager.enqueue(request)

        registerReceiver(onDownloadComplete,
IntentFilter(DownloadManager.ACTION_DOWNLOAD_COMPLETE))
  }

  private val onDownloadComplete: BroadcastReceiver = object : BroadcastReceiver() {
    override fun onReceive(context: Context?, intent: Intent?) {
      val id = intent?.getLongExtra(DownloadManager.EXTRA_DOWNLOAD_ID, -1)
      if (id == downloadId) {
        Toast.makeText(this@ForegroundService, "Download Completed",
Toast.LENGTH_SHORT).show()
        updateNotification("Download Complete")
      }
    }
  }
  private fun updateNotification(contentText: String) {
    val notification = NotificationCompat.Builder(this, CHANNEL_ID)
      .setContentTitle("Foreground Service")
      .setContentText(contentText)
      .setSmallIcon(R.drawable.img)
      .build()
    startForeground(1, notification)
  }
  private fun createNotificationChannel() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
      val serviceChannel = NotificationChannel(
        CHANNEL_ID,
        "Foreground Service Channel",
        NotificationManager.IMPORTANCE_DEFAULT
      )
      val manager = getSystemService(NotificationManager::class.java)
      manager.createNotificationChannel(serviceChannel)
    }
  }
  override fun onDestroy() {
    super.onDestroy()
    mediaPlayer.stop()
    unregisterReceiver(onDownloadComplete)
  }
  override fun onBind(intent: Intent?): IBinder? {
    return null
  }
}
```

AndroidManifest.xml Permissions

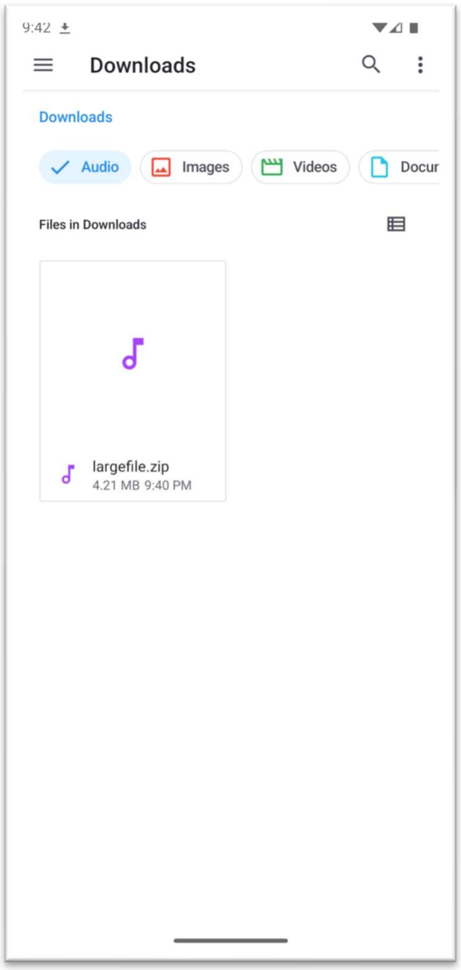
```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
    android:maxSdkVersion="28" />
<uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
```

Output Screenshots

App Interface:



Downloaded File:



Result

Thus, an Android application was developed that utilized a foreground service to concurrently play a video and download a file, and the implementation was verified successfully.