### 1. 在onStartCommand方法中将flag设置为START\_STICKY;

return Service.START\_STICKY;

### 2. 在xml中设置了android:priority

<!--设置服务的优先级为MAX\_VALUE-->  
 <service android:name=".MyService"  
 android:priority="2147483647"  
 >  
 </service>

### 3. 在onStartCommand方法中设置为前台进程

@Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 Notification notification = new Notification(R.mipmap.ic\_launcher, "服务正在运行",System.currentTimeMillis());  
 Intent notificationIntent = new Intent(this, MainActivity.class);  
 PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,notificationIntent,0);  
 RemoteViews remoteView = new RemoteViews(this.getPackageName(),R.layout.notification);  
 remoteView.setImageViewResource(R.id.image, R.mipmap.ic\_launcher);  
 remoteView.setTextViewText(R.id.text , "Hello,this message is in a custom expanded view");  
 notification.contentView = remoteView;  
 notification.contentIntent = pendingIntent;  
 startForeground(1, notification);  
 return Service.START\_STICKY;  
  
 }

### 4. 在onDestroy方法中重启service

@Override  
public void onDestroy() {  
 super.onDestroy();  
 startService(new Intent(this, MyService.class));  
}

### 5. 用AlarmManager.setRepeating(...)方法循环发送闹钟广播,接收的时候调用service的onstart方法

Intent intent = new Intent(MainActivity.this,MyAlarmReciver.class);  
 PendingIntent sender = PendingIntent.getBroadcast( MainActivity.this, 0, intent, 0);  
  
 // We want the alarm to go off 10 seconds from now.  
 Calendar calendar = Calendar.getInstance();  
 calendar.setTimeInMillis(System.currentTimeMillis());  
 calendar.add(Calendar.SECOND, 1);  
 AlarmManager am = (AlarmManager) getSystemService(ALARM\_SERVICE);  
 //重复闹钟  
 /\*\*  
 \* @param type  
 \* @param triggerAtMillis t 闹钟的第一次执行时间，以毫秒为单位  
 \* go off, using the appropriate clock (depending on the alarm type).  
 \* @param intervalMillis 表示两次闹钟执行的间隔时间，也是以毫秒为单位  
 \* of the alarm.  
 \* @param operation 绑定了闹钟的执行动作，比如发送一个广播、给出提示等等  
 \*/  
 am.setRepeating(AlarmManager.RTC\_WAKEUP, calendar.getTimeInMillis(), 2 \* 1000, sender);

### 6. 目前市场面的很多三方的消息推送SDK唤醒APP,例如Jpush



img

### **总结**

这纯粹是面试的时候忽悠一下面试官，不代表着你的Service就永生不死了，只能说是提高了进程的优先级。迄今为止我没有发现能够通过常规方法达到流氓需求(通过长按home键清除都清除不掉)的方法，目前所有方法都是指通过Android的内存回收机制和普通的第三方内存清除等手段后仍然保持运行的方法，有些手机厂商把这些知名的app放入了自己的白名单中，保证了进程不死来提高用户体验（如微信、QQ、陌陌都在小米的白名单中）。如果从白名单中移除，他们终究还是和普通app一样躲避不了被杀的命运。