Hiểu về launchMode của Activity:

standard, singleTop, singleTask và singleInstance

Chỉ định 1 launchMode

Về cơ bản ta có thể chỉ định 1 launchMode vào thẻ <activity> trong file AdroidManifest.xml:

<**activity**

**android:name=".SingleTaskActivity"**

**android:label="singleTask launchMode"**

**android:launchMode="singleTask"**>

Có 4 loại launchMode:

**Standard**

Đây là chế độ mặc định.

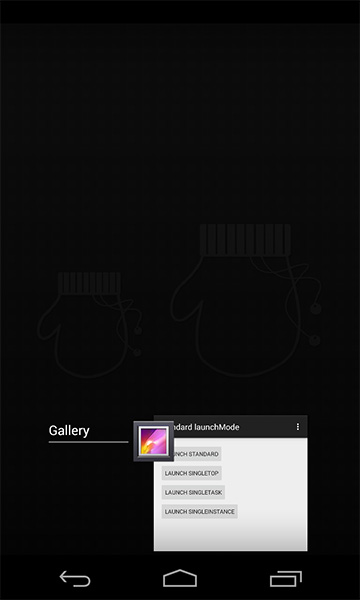
Nếu Activity được đặt mode này thì sẽ luôn có 1 activity mới được tạo ra để làm việc cùng với Intent đã được gửi đi. Activity có thể được khởi tạo nhiều lần, mỗi thể hiện có thể thuộc về các task khác nhau và 1 task có nhiều thể hiện.

**Cách hoạt động ở phiên bản trước Android Lollipop**

Ở đây Activity sẽ được tạo và đặt trên đỉnh Stack trong cùng 1 Task khi đã gửi 1 Intent

Hình ảnh dưới đây cho biết điều gì sẽ xảy ra khi ta share ảnh cho 1 standard Activity. Activity đó sẽ được xếp chồng lên cùng task như mô tả. Mặc dù 2 acitivity đó từ 2 ứng dụng khác nhau

Và đây là những gì bạn nhìn thấy trong Task Manager



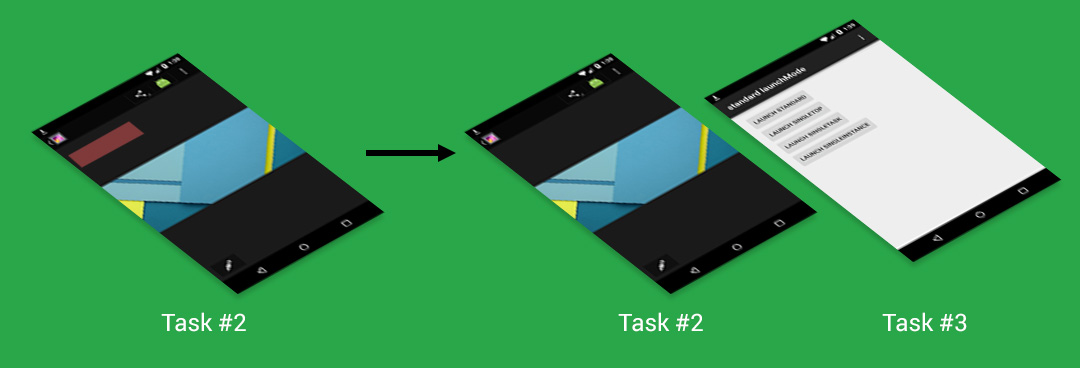
Nếu ta chuyển ứng dụng sang 1 ứng dụng khác và sau đó ta chuyển lại Gallery thì ta vẫn nhìn thấy standard launchMode ở trên đỉnh của Gallery task. Như thế, nếu ta cần làm điều gì đó với Gallery thì ta phải finish Activity vừa được tạo thêm.

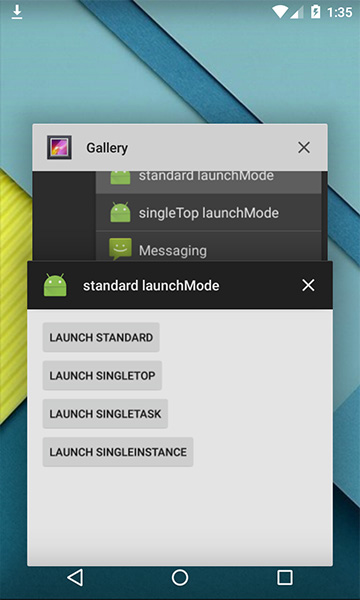
**Cách hoạt động ở phiên bản Android Lollipop**

Nếu các activity trong cùng 1 ứng dụng thì sẽ làm giống như các phiên bản trước android Lollipop, xếp chồng lên đỉnh của task

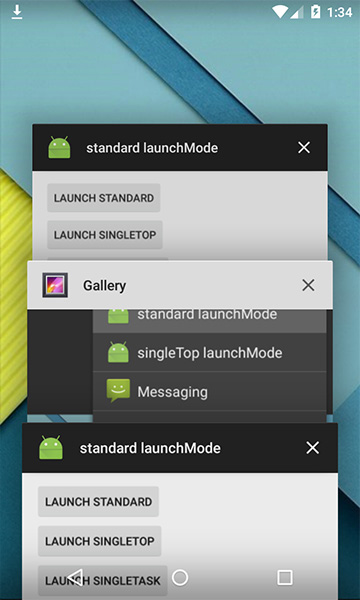


Nhưng trong trường hợp Intent được gửi từ 1 ứng dụng khác thì Task mới được tạo và Activity vừa được tạo mới đó sẽ được đặt làm root Activity như dưới đây



Và đây là những gì bạn sẽ thấy trong Task Manager

Điều này xảy ra vì Task Management system bị sửa đổi trong Lollipop tốt hơn và hợp lý hơn. Trong Lollipop bạn chỉ cần quay trở lại Gallery trong task khác. Bạn có thể bắn Intent khác, task mới sẽ được tạo ra để phục vụ Intent giống như cái trước



**SingleTop**

Giống như **Standard** nhưng khác ở chỗ nếu như đã tồn tại 1 Activity cùng loại ở đỉnh Stack trong cùng 1 Task gọi thì sẽ ko có 1 Activity mới nào được tạo ra, thay vào đó Intent sẽ được gửi tới Activity đã tồn tại qua phương thức **onNewIntent()**.



Trong SingleTop mode thì bạn phải xử lý Intent đến trong cả **onCreate()** và **onNewIntent()** để xử lý trong mọi trường hợp.

Trường hợp sử dụng mode này cho Tìm Kiếm. Trong trường hợp tạo search box để dẫn tới SearchActivity và xem kết quả. Tốt nhất cho UX, bình thường ta luôn đặt search box trong trang kết quả tìm kiếm để cho phép user tìm kiếm mà không có nút Back.

Nếu ta bắn SearchActivity mới để phục vụ việc tìm kiếm thì có 10 Activity cho 10 lần tìm kiếm và sẽ khá khó khăn trong việc muốn quay lại root Activity.

Thay vào đó nếu SearchActivity ở trên đỉnh Stack, tốt hơn hết ta gửi Intent tới Activity đã tồn tại và chỉ cần update kết quả tìm kiếm. Bây giờ sẽ chỉ có 1 SearchActivity trên đỉnh Stack và bạn chỉ cần đơn giản ấn Back để trở về Activity trước.

Tuy nhiên singleTop chỉ làm việc với cùng 1 Task.

Trong trường hợp Intent được gửi đi từ 1 ứng dụng khác cho 1 singleTop Activity thì Activity mới sẽ được bắn như standard launchMode (trước Lollipop: đặt trên đỉnh Stack gọi, trong phiên bản Lollipop: 1 Task mới sẽ được tạo ra)

**SingleTask**

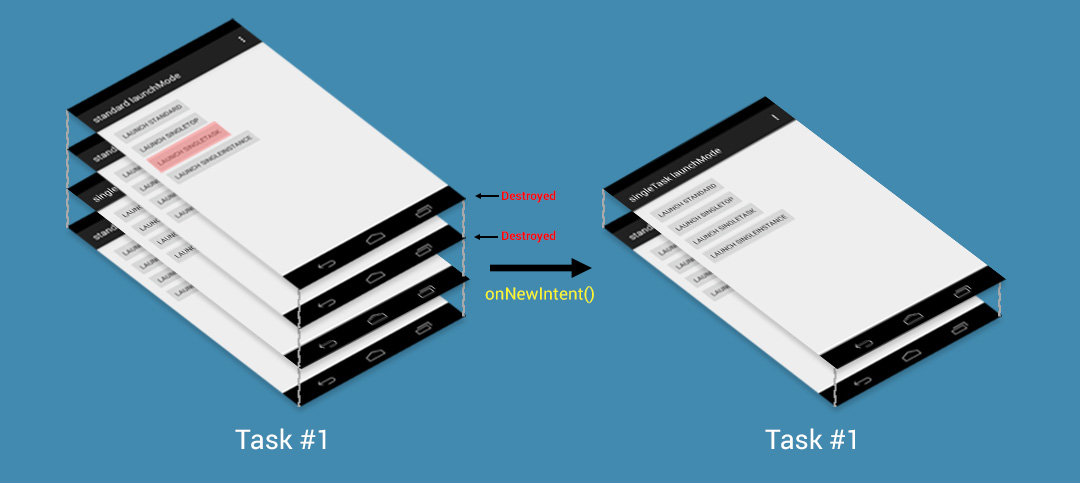
1 Activity singleTask chỉ cho phép có 1 thể hiện trong hệ thống. Nếu đã tồn tại 1 thể hiện Activity trong system thì toàn bộ Task tổ chức các thể hiện sẽ được chuyển lên đỉnh trong khi Intent sẽ được thực hiện thông qua **onNewIntent().** Ngược lại Activity mới sẽ được tạo ra và đặt vào Task riêng biệt.

**Làm việc trong cùng 1 ứng dụng**

Nếu không tồn tại singleTask Activity trong hệ thống thì 1 Activity sẽ được tạo và đặt trên đỉnh Stack trên cùng 1 Task



Nhưng nếu đã tồn tại 1 singleTask Acitivity thì tất cả Acitvity đặt phía trên singleTask Activity sẽ tự động bị hủy để cho Activity xuất hiện trên đỉnh Stack. Intent sẽ được gửi vào **onNewIntent()** của singleTask.



Nhưng từ kinh nghiệm thực tế thì nó không hoạt động như mô tả. SingleTask Activity vẫn xếp trên đỉnh của Task Activity. Ta có thể nhìn trong dumpsys activity để rõ

Task id #239

TaskRecord{428efe30 #239 A=com.thecheesefactory.lab.launchmode U=0 sz=2}

Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x10000000 cmp=com.thecheesefactory.lab.launchmode/.StandardActivity }

Hist #1: ActivityRecord{429a88d0 u0 com.thecheesefactory.lab.launchmode/.SingleTaskActivity t239}

Intent { cmp=com.thecheesefactory.lab.launchmode/.SingleTaskActivity }

ProcessRecord{42243130 18965:com.thecheesefactory.lab.launchmode/u0a123}

Hist #0: ActivityRecord{425fec98 u0 com.thecheesefactory.lab.launchmode/.StandardActivity t239}

Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x10000000 cmp=com.thecheesefactory.lab.launchmode/.StandardActivity }

ProcessRecord{42243130 18965:com.thecheesefactory.lab.launchmode/u0a123}

Nếu bạn muốn singleTask Activity như mô tả trong document: tạo task mới và đặt làm root Activity. Bạn cần khai báo **TaskAffinity**  như sau:

<**activity**

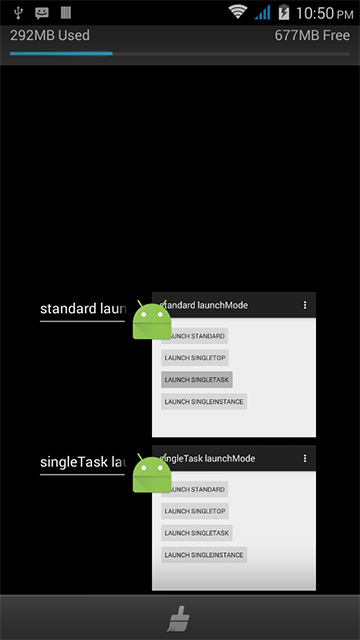
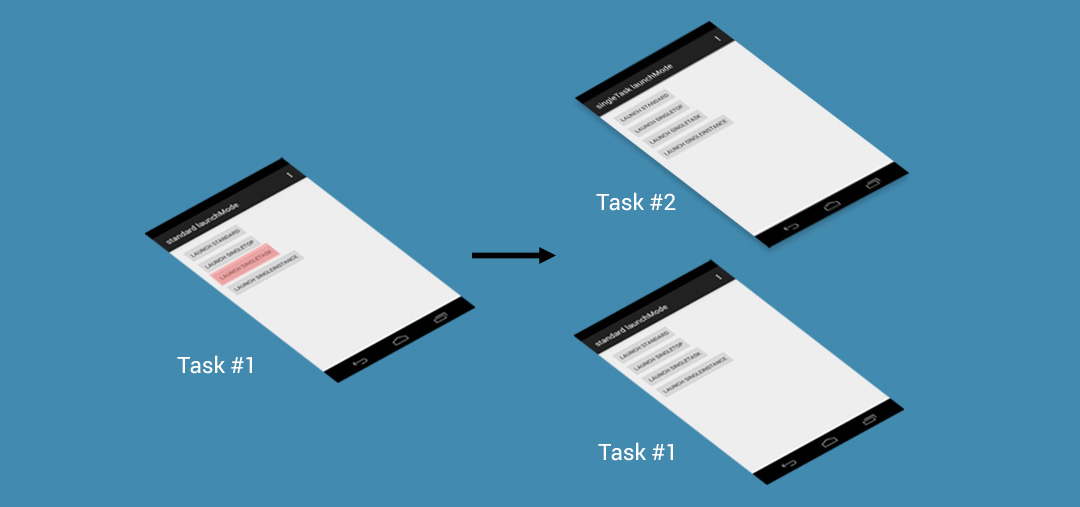
**android:name=".SingleTaskActivity"**

**android:label="singleTask launchMode"**

**android:launchMode="singleTask"**

**android:taskAffinity=""**>

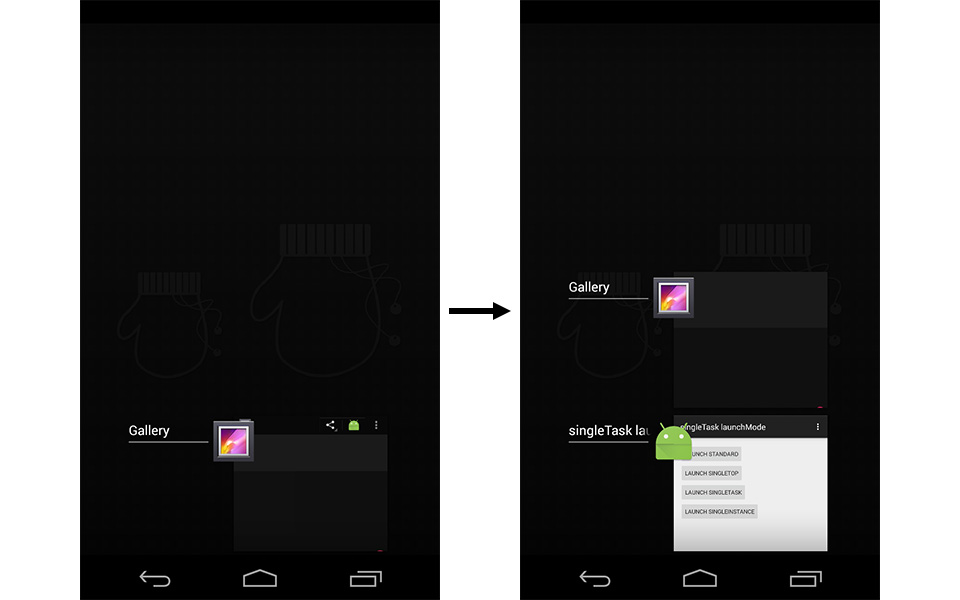
và đây là kết quả



**Liên kết với ứng dụng khác**

Khi Intent được gửi từ ứng dụng khác và không tồn tại Activity nào trong hệ thống thì Task mới sẽ được tạo và tạo mới Acitivity làm root.

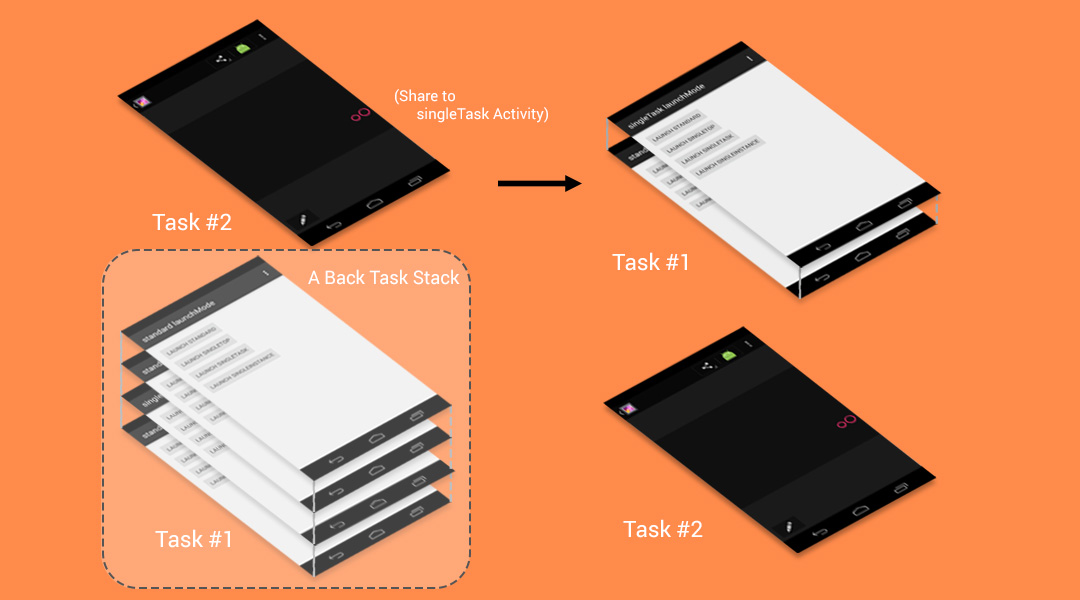




Trừ khi 1 task của singleTask Activity đã tồn tại thì Activity mới sẽ được tạo ra và đặt trên đỉnh.



Trong trường hợp đã tồn tại thể hiện Activity trong bất kì Task nào thì cả trong task sẽ được chuyển lên đỉnh và tất cả Acitivity đặt trên singleTask Activity sẽ bị hủy. Nếu nút Back được ấn thì user sẽ đi qua các Activity trong Stack trước khi trở về Caller Task.



Trường hợp sử dụng cho mode này là Entry Point Activity như: Email Client's Inbox hay Social Network's Timeline. Những Activity này không được thiết kế cho việc có nhiều hơn 1 thể hiện vậy nên singleTask làm điều đó khá hoàn hảo. Tuy nhiên phải dùng mode này cẩn thận vì có

nhiều Activity có thể bị hủy mà không có sự chấp nhận của user.

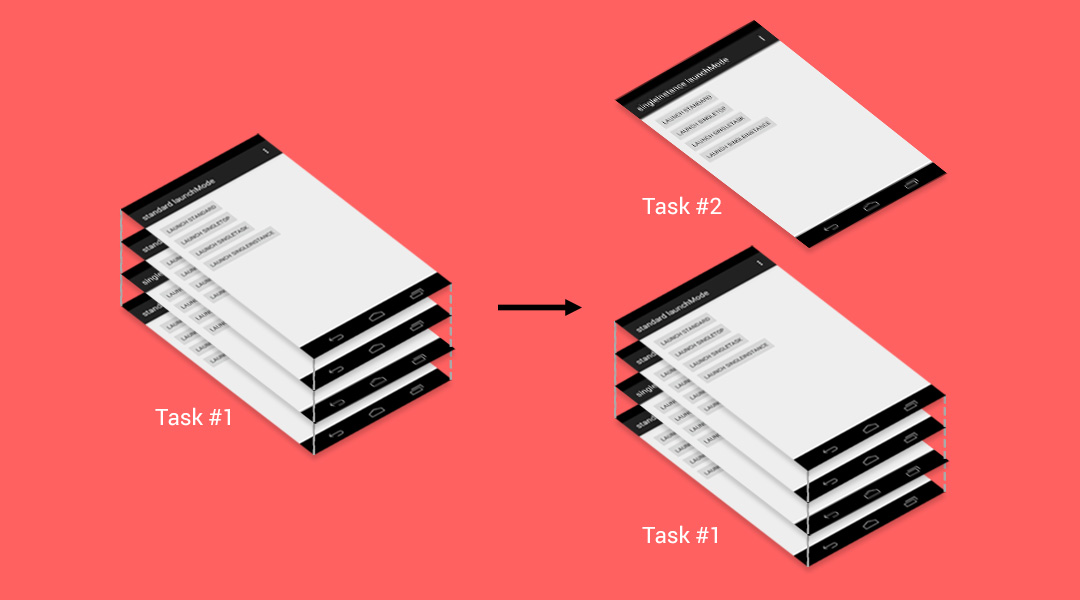
**SingleInstance**

Mode này khá giống với SingleTask nhưng sẽ chỉ có 1 thể hiện của Activity được phép tồn tại trong system. Task chỉ quản lý 1 Activity. Nếu Activity khác gọi singleInstance Activity thì task mới sẽ được tạo ra và đặt Activity mới vào đó.

Tuy nhiên kết quả khá là khác. Từ information trong dumpsys cho thấy có 2 Task trong system nhưng chỉ có 1 thể hiện trong Task Manager (phụ thuộc vào cái ở trên cùng) .

Mặc dù có 1 Task vẫn chạy ỏ background nhưng ta không thể chuyển về foreground.

Đó là điều sẽ xảy ra khi singleInstance Activity được gọi trong khi vẫn còn Activity tồn tại trong Stack.



Đây là trong Task Manager



Task này chỉ có thể có 1 Activity nên ta không thể switch back về Task #1. Chỉ có cách làm điều đó là relaunch ứng dụng từ launcher nhưng nó sẽ khiến singleInstance Task sẽ bị ẩn dưới background.

Tuy nhiên có cách để giải quyết vấn đề này là chỉ định **TaskAffinity** cho singleInstance Activity để cho phép multiple Task trên Task Manager

<**activity**

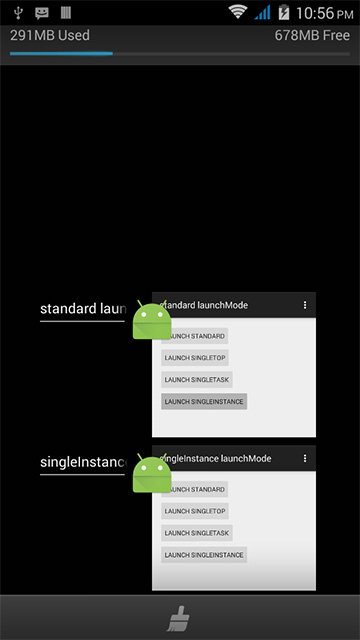
**android:name=".SingleInstanceActivity"**

**android:label="singleInstance launchMode"**

**android:launchMode="singleInstance"**

**android:taskAffinity=""**>

Đây là kết quả



**Intent Flags**

Có 1 cách khác không cần khai báo trong AndroidManifest.xml

là set Flag

Intent intent = **new** Intent(StandardActivity.**this**, StandardActivity.**class**);

intent.addFlags(Intent.***FLAG\_ACTIVITY\_SINGLE\_TOP***);

startActivity(intent);