

# Hidden activities are not destroyed under memory pressure

May 23, 2017 by [Gil Vegliach](#)

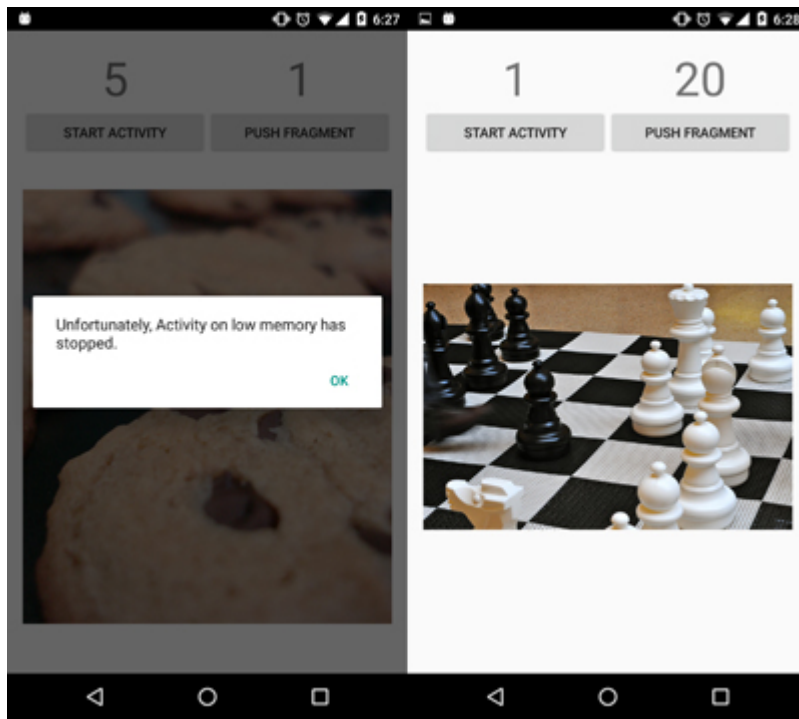
Many developers after reading the official documentation believe that the system will reap the hidden activities of a task when running out of memory. In this post we will see this is not the case and how to solve the problem.

Let's take a look at the [official docs](#):

*"If an activity is paused or stopped, the system can drop it from memory either by asking it to finish (calling its `finish()` method), or simply killing its process. When the activity is opened again (after being finished or killed), it must be created all over."*

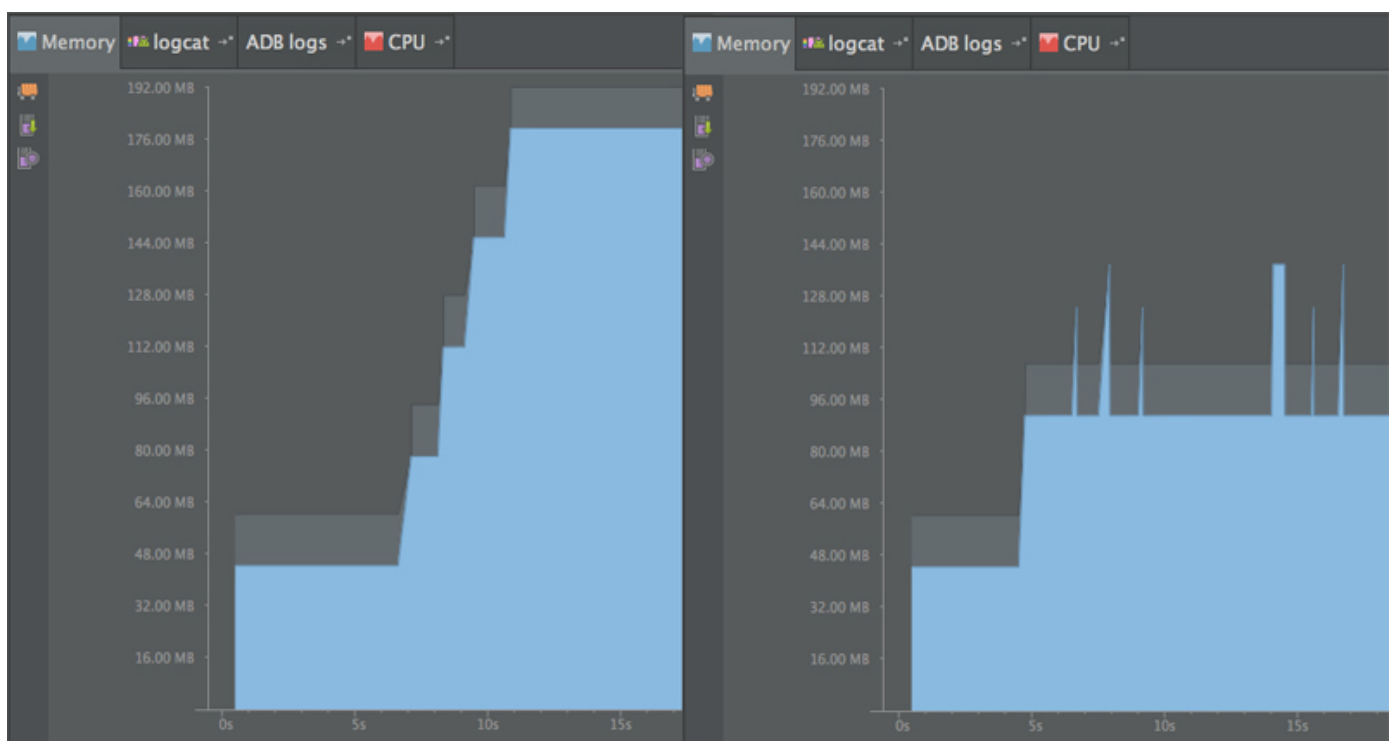
This simply does not happen. According to [this stackoverflow answer](#) by Dianne Hackborn, one of the core engineer on the Android platform, *"the only memory management that impacts activity lifecycle is the global memory management across all processes"*, basically the out-of-memory killer killing entire processes to reclaim memory. This means in particular that `onDestroy()` won't be called on your activities and that your application will blow up if you keep too many memory-consuming activities in a task.

How to solve the problem? Well, still in the same answer, Dianne Hackborn says to use fragments as they destroy their views when they get to the backstack. To prove this point I coded a [simple project](#) (clone from [here](#), then import `ActivityLowMemory` from the Android folder).



At the start, there is an activity with a single fragment and the fragment loads an expensive bitmap. There are two buttons with two relative counters: how many activities are in the task and how many fragments are in the current activity. If you start launching activities you will see that you run out of memory pretty quickly (at the sixth activity on my Nexus 5X). Instead, if you push fragments, you won't blow up because `onDestroyView()` will be called and the garbage collector will be able to collect the expensive bitmap.

You can check the memory consumption with the Memory View of Android Studio, as seen in the next picture, activities on the left and fragments on the right. Note there is a bit of jag because images are loaded on the UI thread, which is usually bad practice but not the point of this sample.



So can activities ever be “killed”? Actually if the process is killed by the system, or by your main thread blowing up, the information in the activity task is preserved. That is, Android will remember which activities were launched or, more precisely, it will remember all the activities that had saved their state so far. When you resume the task, `onCreate()` is called only on the topmost activity, making the other underlying activities effectively “killed”. When you press back and an old activity is to be seen, its `onCreate()` is finally called. You can verify this checking the logs in the [sample project](#).

Finally, an [official bug](#) has been filed to fix the documentation, but hasn’t been worked on so far. You can star it to raise its priority so that we will all get better documentation.

Categories: [Android](#)

[←Git: Committed on master instead of forking a branch](#)

[Retrofit 2: Code walkthrough →](#)

0 Comments

clevercoder.net

 Login ▾ Recommend 4 Tweet Share

Sort by Best ▾



Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Be the first to comment.

ALSO ON CLEVERCODER.NET

### Toptal: How I passed the interview

52 comments • 2 years ago

**Kenny** — Was the live coding round Android specific?

### Three tools to stay in the zone

4 comments • 3 years ago

**Gil Vegliach** — Thanks for i3wm, it is really cool! Take a look also to tmux, a terminal multiplexer very popular with ruby devs. It is

### Booking.com: How I passed the interview

23 comments • 2 years ago

**Aakash Doshi** — Thanks for sharing. Can you please provide sample questions?

### Pro Tip: Android Studio shortcuts for unit tests

2 comments • 2 years ago

**Gil Vegliach** — Good idea! Names added in parentheses.

 Subscribe  Add Disqus to your site  Add Disqus  Disqus' Privacy Policy  Privacy  Privacy

## Newsletter Sign-up

## Share



## Recent Posts

[Clean Architecture by Uncle Bob: Summary and review](#)

[Bash script: Change your MAC address](#)

[Zalando: How to pass the interview](#)

[AWS Certified Solutions Architect: How I got certified](#)

[AWS Certification: Half-way update](#)

## Archives

[September 2018](#)

[May 2018](#)

[April 2018](#)

[February 2018](#)

[November 2017](#)

[October 2017](#)

[September 2017](#)

[August 2017](#)

[July 2017](#)

[June 2017](#)

[May 2017](#)

[April 2017](#)

[March 2017](#)

[February 2017](#)

[January 2017](#)

[December 2016](#)

[November 2016](#)

[October 2016](#)

[April 2016](#)

© 2018 CleverCoder.net