Assignment 4: System Service and Binder Tracing

The goal of this assignment is to get you familiarized with Binder. This assignment will require you to explore both the framework and the kernel. You will add a new system service and instrument the kernel as well as the framework to trace the registration and call process. You will also need to compile the kernel with the instrumented Binder driver as well as the framework with the new service.

Deadline for this assignment is 10/14/2013.

Adding a New System Service

The first part of the assignment is to add a new system service. This requires you to write a new service, expose the interface, add it to the system server, and write an app to call your new service.

1. As explained in the lecture, the system server adds and runs most of the system services. You need to look at that source file and add your own service.
2. You also need to write an AIDL file to expose your interface and also extend .Stub to provide the actual implementation. A good tutorial for this is at: http://developer.android.com/guide/components/aidl.html
    a. This part can be a bit tricky if it's the first time using AIDL.
3. Your new system service should have one method that returns the current time, e.g., you can name that method getTime().
4. Then you need to add your service to the build system. A good starting point to look at how to add a new AIDL file as part of the system server is:
    a. <platform root>/frameworks/base/Android.mk.
    b. You need to be able to compile your new service with the rest of the framework.
5. The third step is to make actual calls to getTime().
    a. For this, call getTime() inside of Launcher.onClick() so you can log the time at each app start using Launcher.
    b. This part can be a bit tricky because you need to figure out the proper use of the interface. But you can use either Context.getSystemService() (which needs some more framework changes) or ServiceManager.getService() to get the service remote object.

Instrumenting Binder

We will now track the registration and calling of your service in Binder. We will instrument both the framework and the kernel for this.

1. The first step is to instrument the framework.
    a. You need to instrument the Binder context manager so that when your new

service gets registered, you can log the registration. We suggest that you use the string name of your service to identify it in the context manager.

   b. You also need to instrument every lookup of your service.

2. The second step is to instrument the kernel.
   a. You need to instrument the Binder driver to identify all registration/lookup/calls to your service.
   b. You can log this using printk() in the kernel and see it using dmesg.