군산대학교
KUNSAN NATIONAL UNIVERSITY

NDK

모바일 시스템 SW

남 광 우

Java JNI(Java Native Interface)

❖ JNI란

- ✓ JVM에서 돌아가는 Java 코드와 C/C++로 구현된 코드가 상호참조하기위해 사용하는 programming framework
- ✓ Platform-dependent한 라이브러리를 사용하고자 할 때, 혹은 기존의 프로그램을 Java에서 접근가능하도록 변경하고자 할 때 쓰임
- ✓ Java와 C가 하나의 object를 참조할 수 있다

설치 : Android NDK

❖ Android NDK 다운로드

- ✓ http://developer.android.com/tools/sdk/ndk/index.html
- ✓ Download android-ndk-r8c-windows.zip

❖ 설치 방법

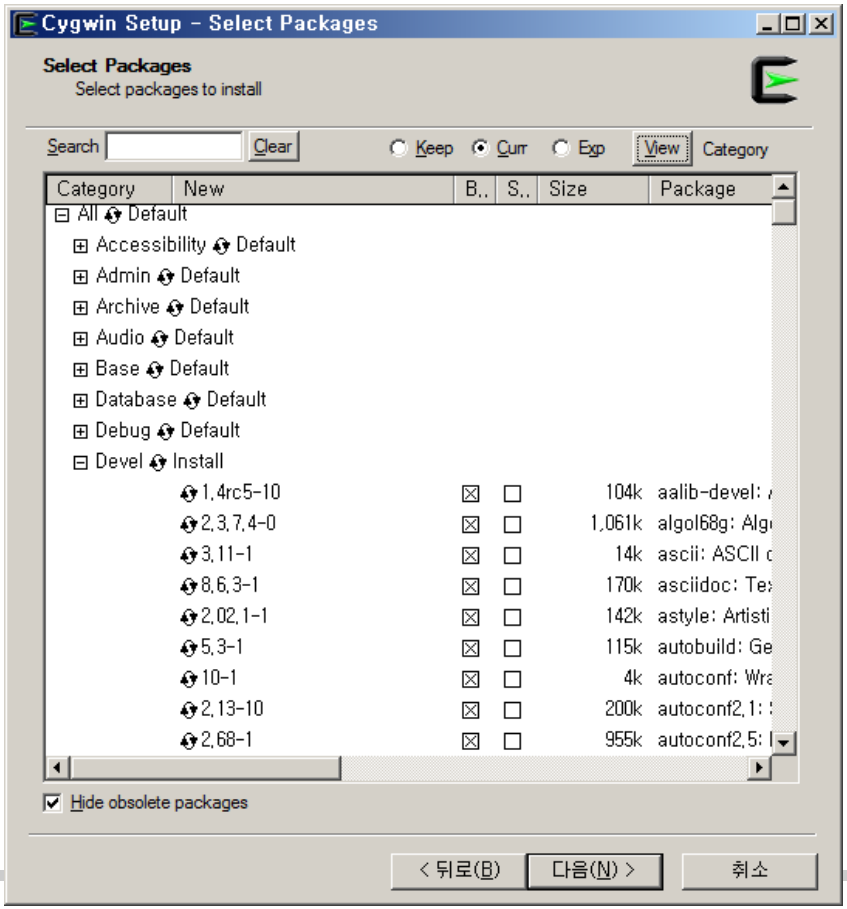
- ✓ zip 파일을 적절한 곳에 옮긴후 unzip

설치 : Cygwin의 설치

- **Cygwin이란?**
  - ✓ Windows에서 Unix like tool 및 개발 환경을 지원하는 툴

- **다운로드**
  - ✓ http://www.cygwin.com

필수 설치
: gcc, make, vim

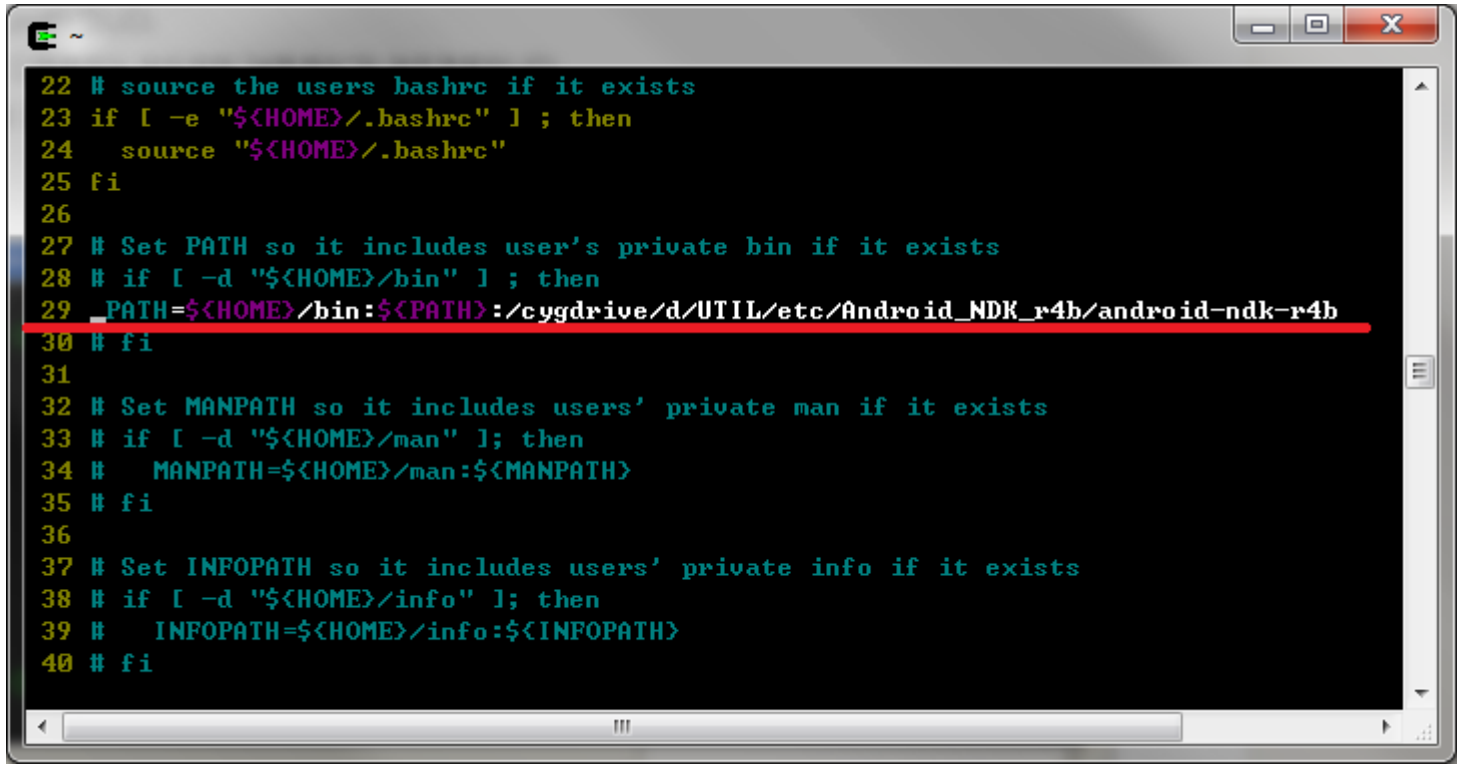


설치 : Cygwin의 설치

BASH

❖ Cygwin에서 path의 수정

✓ ndk-build 명령을 수행하기 위한 path를 .bash_profile 파일의 PATH에 추가



/cygdrive => windows의 내컴퓨터 디렉토리임

Android NDK란?

- **A toolset that lets you embed in you app native source code**
- **C, C++(recently supported December 2010) and assembly(?)**
- **It is supported on android cupcake(1.5)+**
- **It is aimed to**
  - ✓ Bring native libraries in android (code reusability)
  - ✓ Make some parts of the application really fast using code generated for arm-like cpus
- **Most of the time android SDK is prerequisite for NDK**
- **Under heavy development**

Android NDK를 언제 쓸것인가?

- 단지 C, C++을 더 쓰고 싶어서라면 쓰지말것..
- NDK 사용의 단점을 뛰어넘는 장점이 있을때만 사용할 것
  - ✓ 응용 개발의 복잡성이 증가하며,
  - ✓ 디버깅이 어려움
  - ✓ 그리고, C, C++을 사용한다고 해서 항상 성능이 빨라지는 것도 아님
- But,
  - ✓ OpenGL 과 같은 특정 라이브러리를 사용하고 싶거나,
  - ✓ 이미 개발된 라이브러리들을 port해서 사용하고 싶거나,
  - ✓ 특정 부분에서의 명확한 성능 향상이 예측될때는 사용해도 좋음

NDK 개발 방법

❖ 두가지 개발 방법

- ✓ Java JNI를 사용해서 NDK에서 제공하는 API에 접근하는 방법
- ✓ NativeActivity class와 native code를 이용해서 구현하는 방법
  - 2.3 이상
  - Java Code 없이 Native Code만으로 개발 가능

NDK 개발 방법의 예

❖ Android OpenGL 프로그래밍의 3가지 방법

- ✓ Java OpenGL
  - Java로 프로그래밍하여 Java 바인딩 된 OpenGL 함수 사용
- ✓ JNI OpenGL
  - Java에서 컨텍스트만 만들고 실제 GL 함수는 C, C++ 함수로 so파일로 만든뒤
  - Java 클래스에서 호출
- ✓ NativeActivity OpenGL
  - Java를 전혀 사용하지 않고 C/C++ 만 이용해서 개발
  - Android-9(2.3)부터 나온 제작방법으로 ndk/samples/native-activity 예제 이용

Java에서 C 호출 하기

- **Library loading in Java**
  - ✓ static { System.loadLibrary(“libraryName”); }

- **Declaration in Java**
  - ✓ public native void MethodName();

- **Implementation in C**

```
void Java_PackageName_ClassName_MethodName
( JNIEnv* env, jobject obj )
{
/*
    구현
*/
}
```

Java에서 C 호출 하기(파라미터가 있는경우)

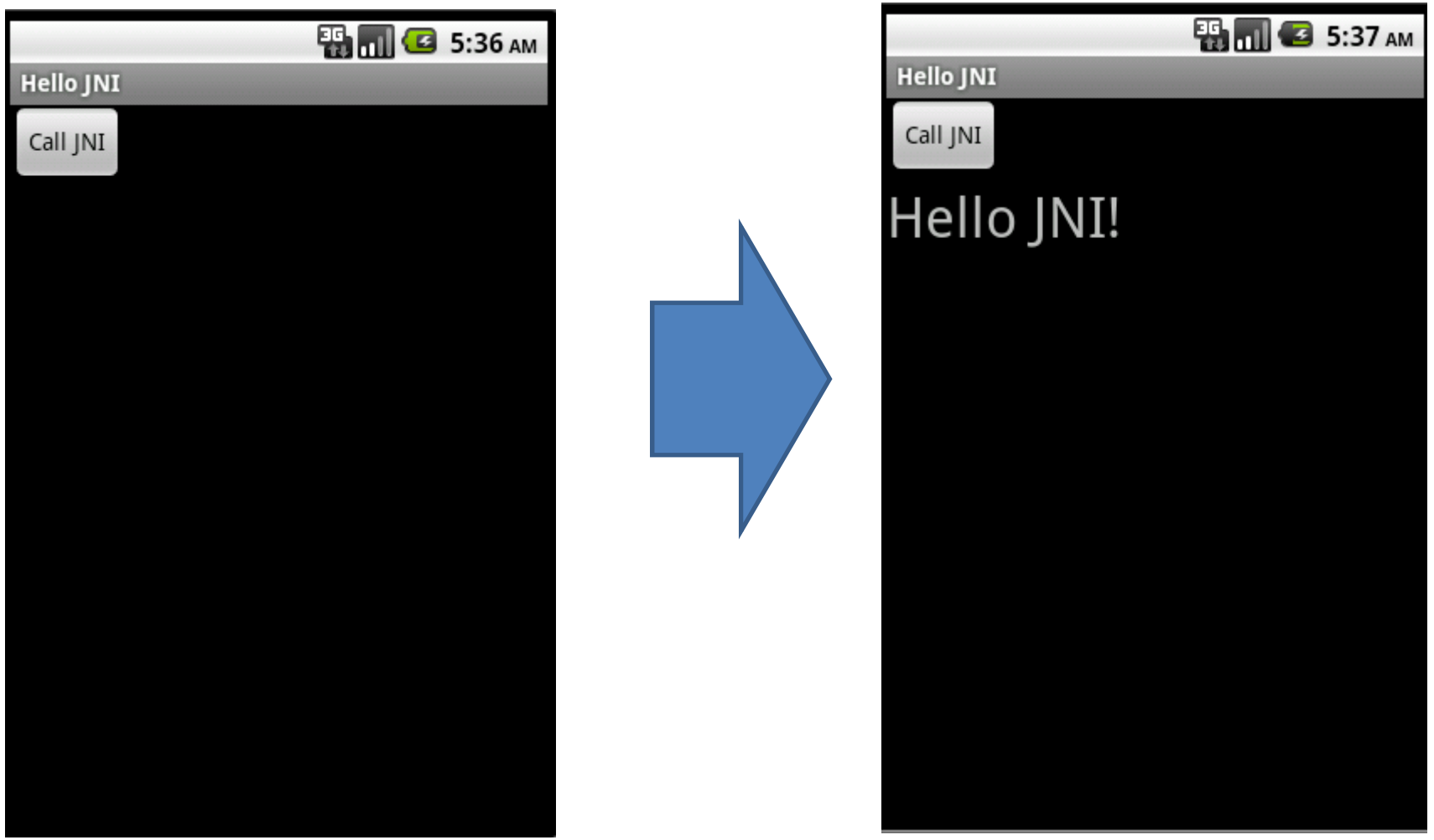
- **Declaration in Java**
  - ✓ public native void MethodParam(int a, String b, float[] c);

- **Implementation in C**

```
void Java_PackageName_ClassName_MethodName
( JNIEnv* env, jobject obj, jint a, jstring b, jfloatArray c)
{
    /* implement Native Method Here */
    jfloat* cArray = (*env)->GetFloatArrayElements(env, c, 0);
}
```

예제 프로그램

- **Java Button 눌러 C Native Call하기**



예제 프로그램

❖ main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   >
   <Button
       android:id="@+id/callbtn"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Call JNI"
   />
       <TextView
              android:id="@+id/rettxt"
       android:layout_width="fill_parent"
       android:layout_height="wrap_content"
       android:textSize="15pt"
   />
</LinearLayout>
```

예제 프로그램

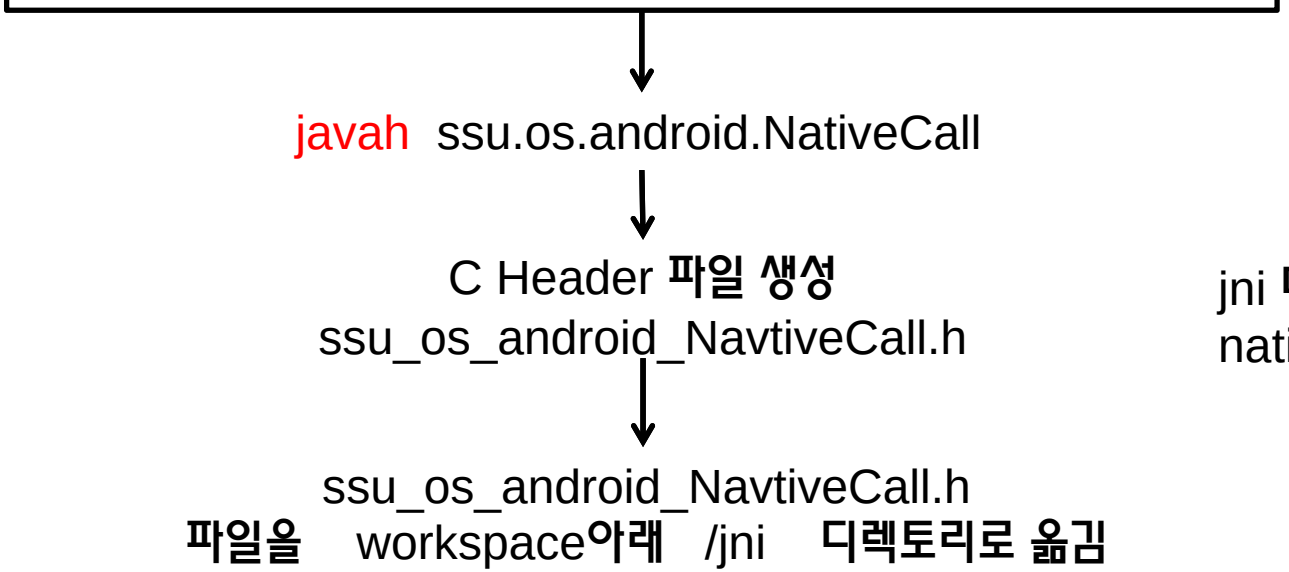
NdkTestActivity.java

```
package ssu.os.android;

public class NdkTestActivity extends Activity {
   /** Called when the activity is first created. */
        private Button bv;
        private TextView tv;

   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);

      bv = (Button)findViewById(R.id.callbtn);
      tv = (TextView)findViewById(R.id.rettxt);

      bv.setOnClickListener(new OnClickListener() {
       public void onClick(View v) {
              NativeCall nativeCall = new NativeCall();
              int ret = nativeCall.add(10, 20);
              String retStr = nativeCall.stringFromJNI();
              Log.i("TAG", retStr + ret);
              tv.setText(retStr);
       }
      });
   }
}
```

예제 프로그램

❖ NativeCall.java

```
package ssu.os.android;

public class NativeCall {
        static {
                System.loadLibrary("my_lib");
        }
        public native String stringFromJNI();
        public native int add(int a, int b);
}
```



jni 디렉토리에
native code 저장

예제 프로그램

❖ mylib.c 파일 구현

- ✓ /jni 디렉토리에 구현

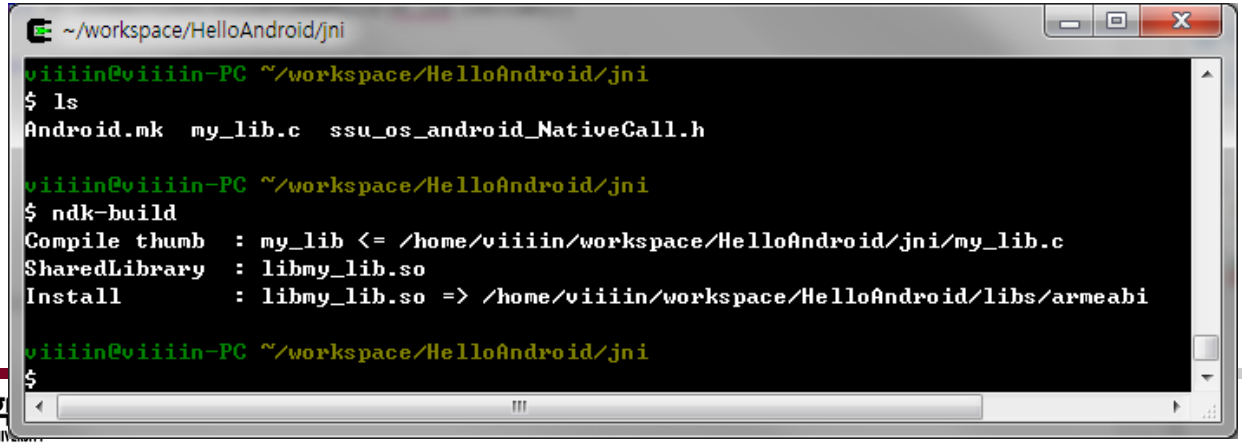
```
#include "ssu_os_android_NativeCall.h"

JNIEXPORT jstring JNICALL Java_ssu_os_android_NativeCall_stringFromJNI(JNIEnv *env, jobject obj)
{
        return (*env)->NewStringUTF(env, "Hello JNI!");
}

JNIEXPORT jint JNICALL Java_ssu_os_android_NativeCall_add(JNIEnv *env, jobject obj, jint a, jint b)
{
        return a + b;
}
```

예제 프로그램

❖ android.mk 파일

```
LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)

LOCAL_MODULE    := mylib
LOCAL_SRC_FILES := mylib.c

include $(BUILD_SHARED_LIBRARY)
```

프로젝트 루트 또는 /jni 디렉토리에서 "ndk-build" 명령을 실행

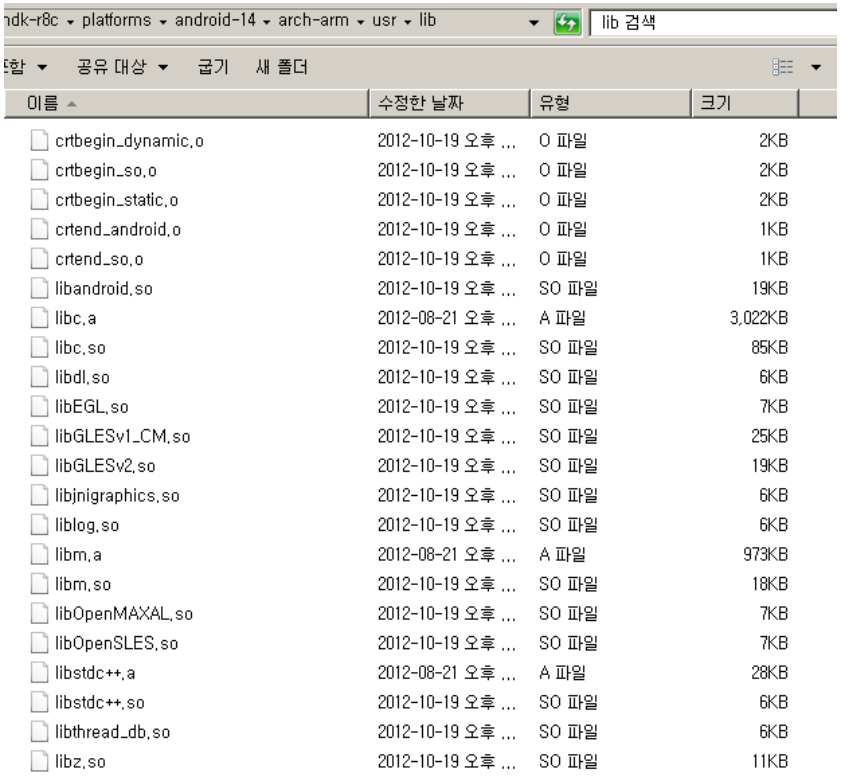


/libs 디렉토리에 libmylib.so 파일이 생성됨

NativeActivity

❖ NativeActivity library

E:\dev\00.bin\android-ndk-r8c\platforms\android-14\arch-arm\usr\lib

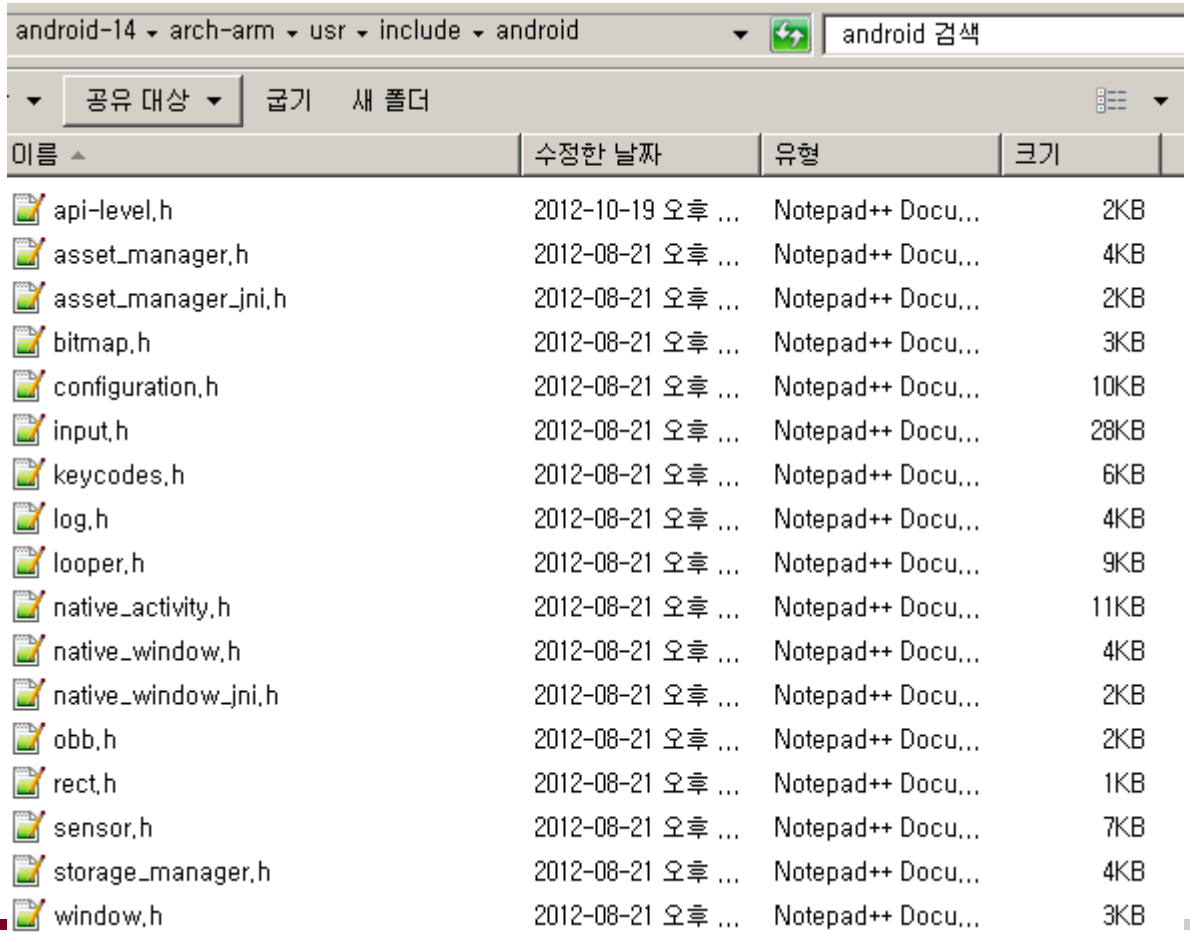


이름	수정한 날짜	유형	크기
crtbegin_dynamic.o	2012-10-19 오후 ...	O 파일	2KB
crtbegin_so.o	2012-10-19 오후 ...	O 파일	2KB
crtbegin_static.o	2012-10-19 오후 ...	O 파일	2KB
crtend_android.o	2012-10-19 오후 ...	O 파일	1KB
crtend_so.o	2012-10-19 오후 ...	O 파일	1KB
libandroid.so	2012-10-19 오후 ...	SO 파일	19KB
libc.a	2012-08-21 오후 ...	A 파일	3,022KB
libc.so	2012-10-19 오후 ...	SO 파일	85KB
libdl.so	2012-10-19 오후 ...	SO 파일	6KB
libEGL.so	2012-10-19 오후 ...	SO 파일	7KB
libGLESv1_CM.so	2012-10-19 오후 ...	SO 파일	25KB
libGLESv2.so	2012-10-19 오후 ...	SO 파일	19KB
libjnigraphics.so	2012-10-19 오후 ...	SO 파일	6KB
liblog.so	2012-10-19 오후 ...	SO 파일	6KB
libm.a	2012-08-21 오후 ...	A 파일	973KB
libm.so	2012-10-19 오후 ...	SO 파일	18KB
libOpenMAXAL.so	2012-10-19 오후 ...	SO 파일	7KB
libOpenSLES.so	2012-10-19 오후 ...	SO 파일	7KB
libstdc++.a	2012-08-21 오후 ...	A 파일	28KB
libstdc++.so	2012-10-19 오후 ...	SO 파일	6KB
libthread_db.so	2012-10-19 오후 ...	SO 파일	6KB
libz.so	2012-10-19 오후 ...	SO 파일	11KB

NativeActivity를 이용한 구현

❖ NDK의 NativeActivity.h

<ndk_root>/platforms/android-9/arch-arm/usr/include/android/native_activity.h



NativeActivity를 이용한 구현

NDK의 NativeActivity.h

```
typedef struct ANativeActivity {

    struct ANativeActivityCallbacks* callbacks;
    JavaVM* vm;
    JNIEnv* env;

    .....
    AAssetManager* assetManager;

} ANativeActivity;
```

NativeActivity

Static

Library

NativeActivity를 이용한 구현

- **NDK의 NativeActivity.h**

```
typedef struct ANativeActivityCallbacks {

    void (*onStart)(ANativeActivity* activity);
    void (*onResume)(ANativeActivity* activity);
    void* (*onSaveInstanceState)(ANativeActivity* activity, size_t* outSize);
    void (*onPause)(ANativeActivity* activity);
    void (*onStop)(ANativeActivity* activity);
    .....

   void (*onLowMemory)(ANativeActivity* activity);
} ANativeActivityCallbacks;
```

NativeActivity를 이용한 구현

- **NDK의 android_native_app_glue.h**

```
struct    android_app {

    void* userData;
    void (*onAppCmd)(struct android_app* app, int32_t cmd);
    ANativeActivity* activity;
    AConfiguration* config;
    ....
}
```

NativeActivity를 이용한 구현

❖ NDK의 android_native_app_glue.c

```
void ANativeActivity_onCreate(ANativeActivity* activity,
        void* savedState, size_t savedStateSize) {
    LOGV("Creating: %p\n", activity);
    activity->callbacks->onDestroy = onDestroy;
    activity->callbacks->onStart = onStart;
    activity->callbacks->onResume = onResume;
    activity->callbacks->onSaveInstanceState = onSaveInstanceState;
    activity->callbacks->onPause = onPause;
    activity->callbacks->onStop = onStop;
    activity->callbacks->onConfigurationChanged = onConfigurationChanged;
    activity->callbacks->onLowMemory = onLowMemory;
    activity->callbacks->onWindowFocusChanged = onWindowFocusChanged;
    activity->callbacks->onNativeWindowCreated = onNativeWindowCreated;
    activity->callbacks->onNativeWindowDestroyed = onNativeWindowDestroyed;
    activity->callbacks->onInputQueueCreated = onInputQueueCreated;
    activity->callbacks->onInputQueueDestroyed = onInputQueueDestroyed;

    activity->instance = android_app_create(activity, savedState, savedStateSize);
}
```

Tracing

❖ **호출되는 순서와 구현할 부분**