

MICROOH 麦可网

Android-从程序员到架构师之路

出品人：Sundy

讲师：高焕堂（台湾）

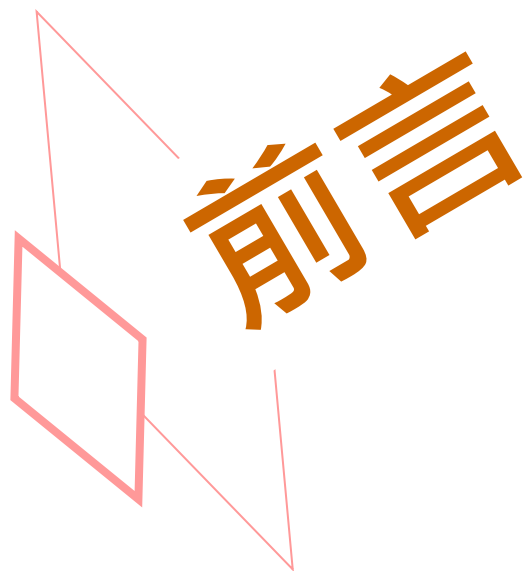
<http://www.microoh.com>

C06_c

JNI：本地线程进入Java层(c)

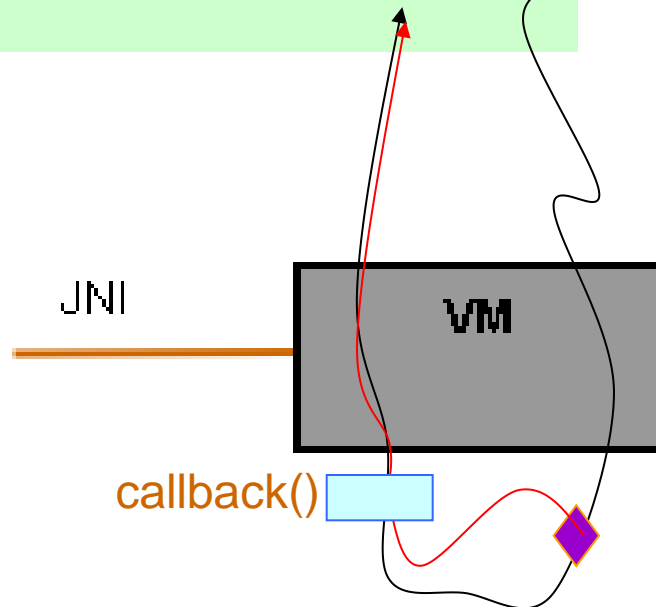
By 高煥堂

3、Native多线程的安全



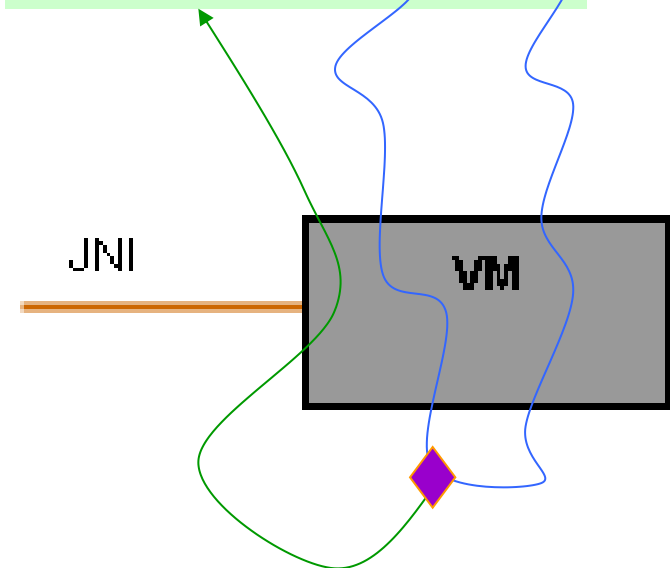
進程(Process)_# 1

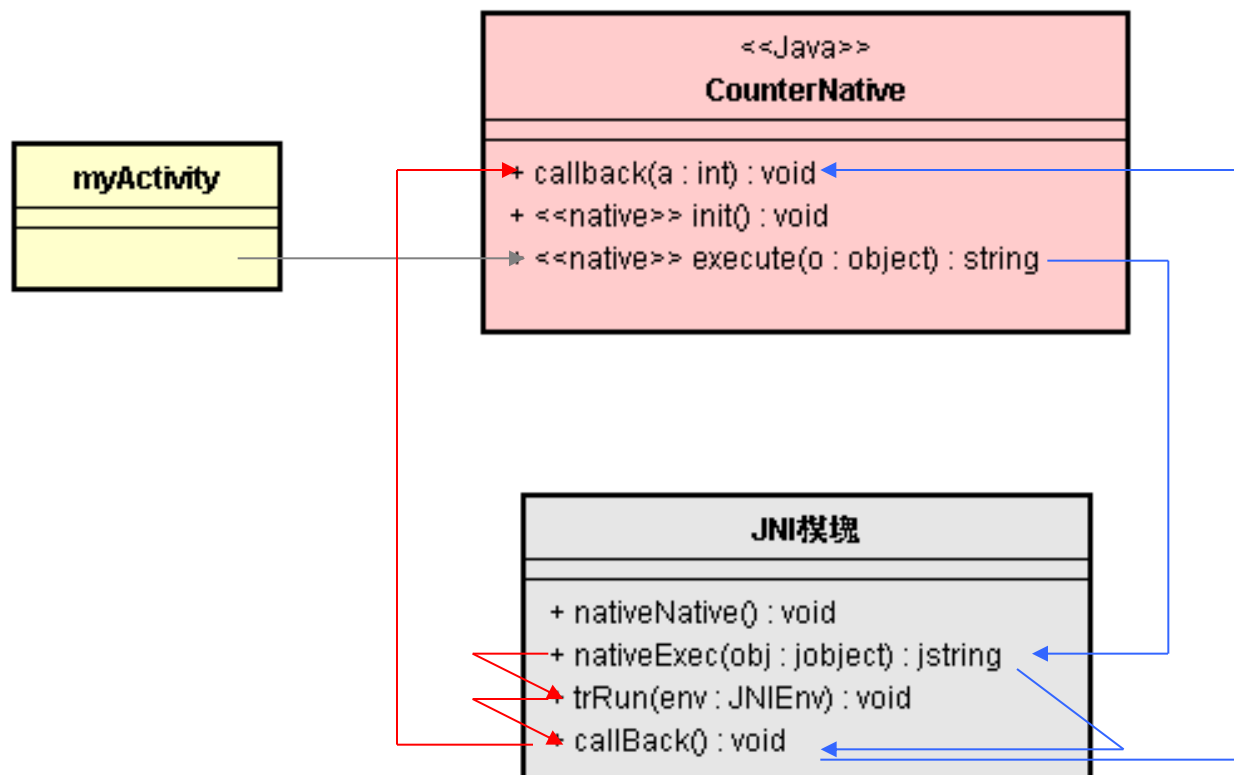
- > 主線程(Main Thread)
- > Message Queue
- > Main Looper



進程(Process)_# 2

- > 主線程(Main Thread)
- > Message Queue
- > Main Looper





- 使用函数pthread_create()函数来诞生Native层的子线程。
- 由于Native函数里执行的线程也能诞生子线程，所以也应该注意其线程安全问题。
例如：

```
/* com_misoo_thread_JTX07.cpp */  
// .....  
JavaVM *gJavaVM;  
int sum;  
pthread_t thread;  
void* trRun( void* );  
void callBack(JNIEnv *);  
jobject mSyncObj;  
//-----  
void Thread_sleep(int t){  
    timespec ts;  ts.tv_sec = t;  
    ts.tv_nsec = 0; nanosleep(&ts, NULL);  
    return;  
}
```



```
void JNICALL
Java_com_misoo_counter_CounterNative_nativeSetup(
    JNIEnv *env, jobject thiz, jobject weak_this){
    jclass clazz = env->GetObjectClass(thiz);
    mClass = (jclass)env->NewGlobalRef(clazz);
    mObject = env->NewGlobalRef(weak_this);
    mid = env->GetStaticMethodID(mClass, "callback","(II)V");
    return;
}
```

```
jstring JNICALL
```

```
Java_com_misoo_counter_CounterNative_nativeExec(
```

```
    JNIEnv *env, jobject thiz, jobject syncObj){
```

```
    mSyncObj = env->NewGlobalRef(syncObj);
```

```
    int t1 = pthread_create( &thread, NULL, trRun, NULL);
```

```
    Thread_sleep(4);
```

```
    callBack(env); // m.t.
```

```
    //-----
```

```
    long pid = getpid();
```

```
    sprintf(sTid, "%lu", pid);
```

```
    jstring ret = env->NewStringUTF(sTid);
```

```
    return ret;
```

```
}
```

```
//-----  
void callBack(JNIEnv *env){  
    env->MonitorEnter(mSyncObj);  
    sum = 0;  
    for(int i = 0; i<=10; i++) {  
        sum += i;  
        Thread_sleep(1);  
    }  
    env->CallStaticVoidMethod(mClass, mid, sum, 666);  
    env->MonitorExit(mSyncObj);  
}
```

```
void* trRun( void* ){
    int status;
    JNIEnv *env;
    bool isAttached = false;

    Thread_sleep(1);
    status = gJavaVM->GetEnv((void **) &env, JNI_VERSION_1_4);
    if(status < 0) {
        status = gJavaVM->AttachCurrentThread(&env, NULL);
        if(status < 0) return NULL;
        isAttached = true;
    }
    callBack(env); // t1
    if(isAttached) gJavaVM->DetachCurrentThread();
    return NULL;
}
```

- 主线程诞生了子线程去执行trRun()函数。
必须先调用

`gJavaVM->AttachCurrentThread(&env, NULL);`

- 才能取得子线程自己所属的JNIEnv对象之
参考了，并且调用Callback()函数。

- 之后，主线程也调用同一Callback函数。
- 于是，在Callback()函数里，使用
`env->MonitorEnter()`和`env->MonitorExit(mSyncObj);`

指令来让各线程能达到同步。

Thanks...



高煥堂