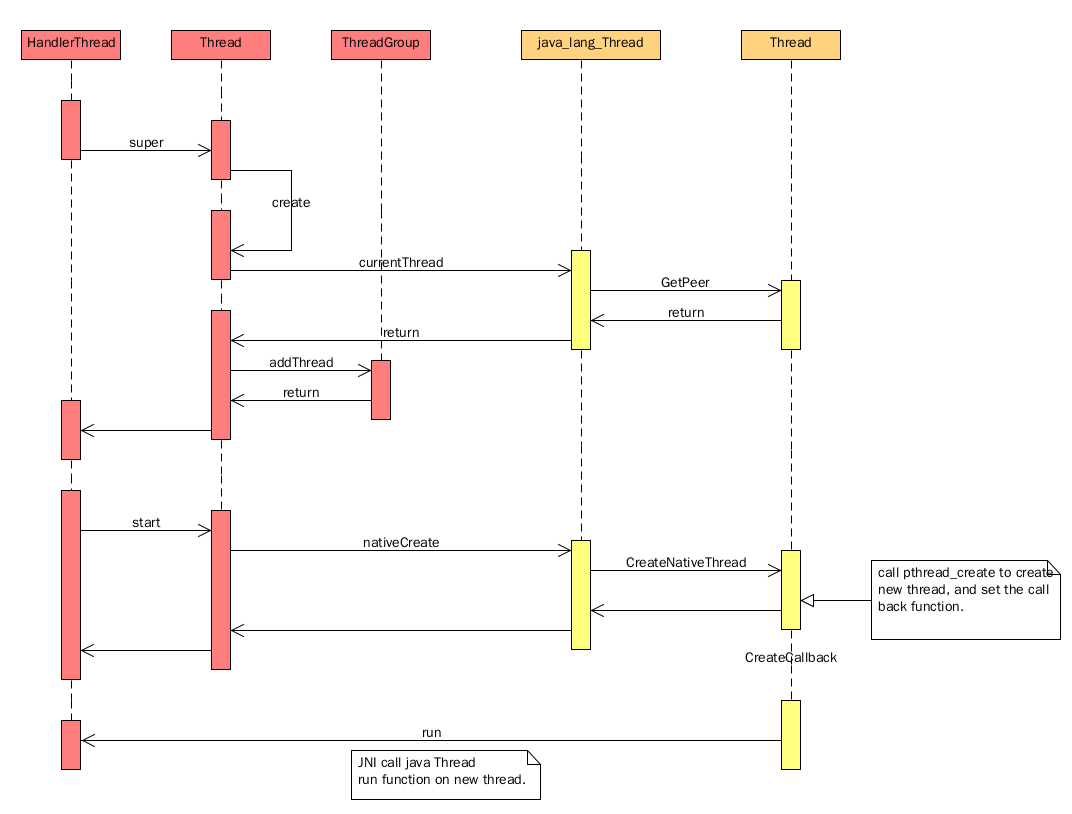
**Looper、handler、Thread**

**1.android 中的Thread**

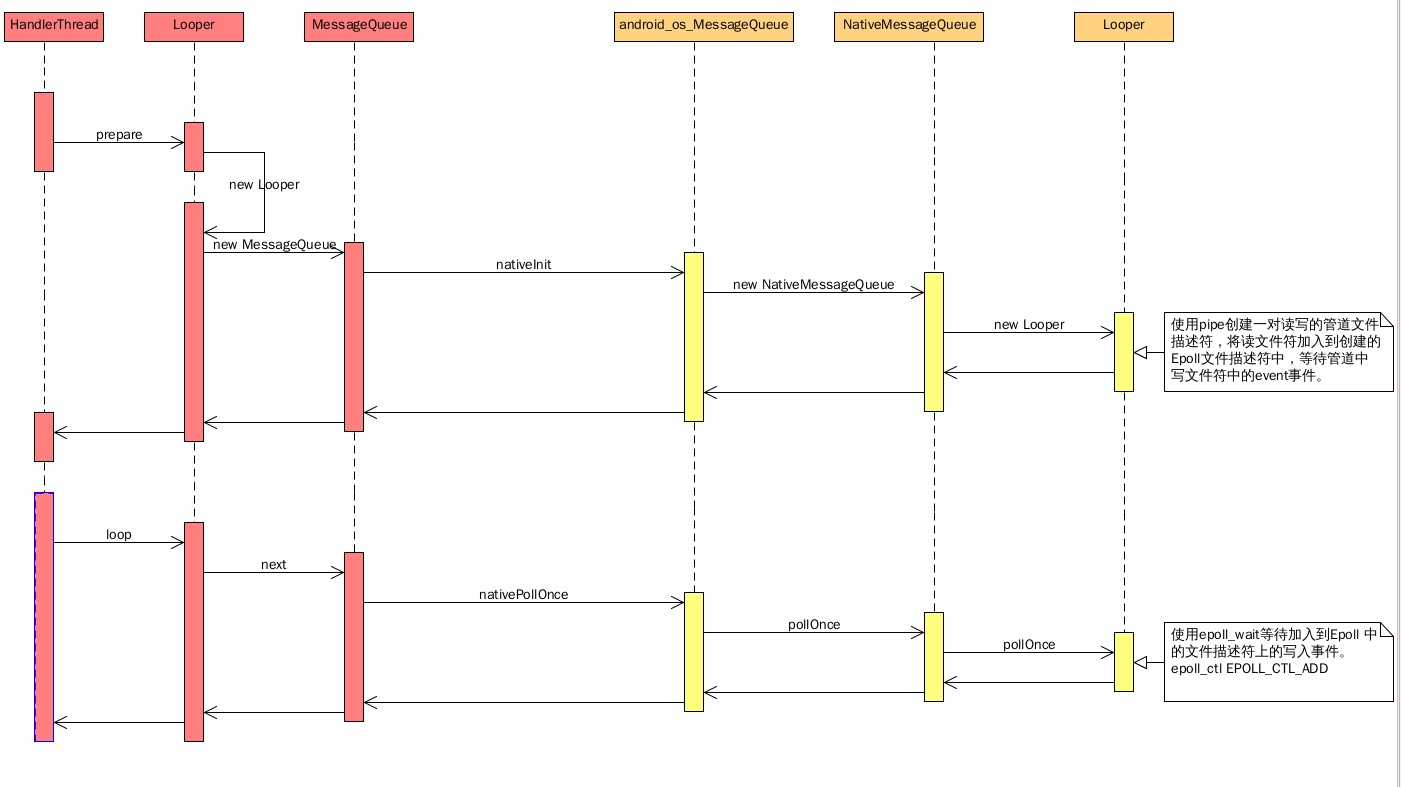
Key class： java.lang.Thread; android.os. HandlerThread;

Call fllow path：android –>Libcore->artRuntime

1.thread 的启动 ****

**2.new thread looper Message的创建和运行**

**Looper时序图：**

****

**发送消息：**

一个looper对应一个Thread，对应一个MessageQueue。

Looperz中使用epoll\_wait的原因：

1.节省cpu。

2.message不会丢失。

线程之间的通信：

AsyncChannel

进程之间发送message：TakeScreenshotService

Handler的创建：

1.使用默认构造函数

public Handler() {

this(null, false);

}

public Handler(Callback callback, boolean async) {

mLooper = Looper.myLooper();

if (mLooper == null) {

throw new RuntimeException(

"Can't create handler inside thread that has not called Looper.prepare()");

}

mQueue = mLooper.mQueue;

mCallback = callback;

mAsynchronous = async;

}

使用当前线程的looper作为该handler的looper。

1. 指定looper来创建相应的handler

public Handler(Looper looper) {

this(looper, null, false);

}

public Handler(Looper looper, Callback callback) {

this(looper, callback, false);

}

public Handler(Looper looper, Callback callback, boolean async) {

mLooper = looper;

mQueue = looper.mQueue;

mCallback = callback;

mAsynchronous = async;

}