在操作系统中，有一些应用程序是以后台服务形式运行的，如Mysql程序等，windows提供服务管理器，可以很方便地启动和停止一个服务。JAVA程序也可以包装成服务程序，像Tomcat那样，通过脚本程序很容易启动和停止tomcat服务。[Apache Common Daemon](http://commons.apache.org/proper/commons-daemon/)可以实现把一个JAVA程序包装成后台服务的功能。停止一个在后台运行的JAVA程序有很多方法，一种简单的方法是找到该程序的PID，直接用kill命令把进程杀死。另一种简单方法是在程序启动后监听一个指定端口，需要停止程序时，通过TCP协议向该端口发送关闭命令即可。示例如下：

* 服务端程序，模拟正在后台运行的JAVA程序

/\*\*

- 本程序模拟一个不断轮询消息队列，从消息队列中取出消息，然后执行业务的程序，

- 程序启动后监听8888端口，当收到“stop”命令时，退出程序

\*/

public class BackgroundApp {

private static boolean forceExit = false;

/\*\*

\* 监听8888端口，当收到“stop”命令时，退出程序

\*/

private static void startMoniter() {

ServerSocket server = null;

BufferedReader br = null;

try {

server = new ServerSocket();

server.bind(new InetSocketAddress("127.0.0.1", 8888));

} catch (Exception e) {

System.out.println("绑定端口失败");

}

try {

while(true){

Socket sock = server.accept();//这里会阻塞，直到收到命令

sock.setSoTimeout(1000); //本地通信设置较短时间

br = new BufferedReader(new InputStreamReader(sock.getInputStream()));

String readContent = br.readLine();

System.out.println("接收到的内容是："+readContent);

// 判断收到信息是否是停止标志

if ("stop".equals(readContent)) {

System.out.println("应用程序准备停止");

forceExit = true;//修改变量值，退出程序

}

br.close();

sock.close();

}

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

private static class MessageHandler implements Runnable{

@Override

public void run() {

while(!forceExit){

try{

//1.从消息队列中取出数据

//2.根据消息内容，执行相应业务

System.out.println("处理业务中...");

Thread.sleep(1000);

}catch (Exception e) {

//donothing

}

}

System.out.println("应用程序停止成功");

System.exit(0);//关闭程序，退出JVM

}

}

public static void main(String[] args) {

Thread workThread = new Thread(new MessageHandler());

workThread.start();//工作线程开始工作

startMoniter();//开启监听器，用于关闭程序

}

}

* 客户端程序，用于向服务端程序发送关闭命令（stop）

public class AppCloseHelper {

public static void main(String[] args) {

String targetHost = "127.0.0.1";

int targetPort = 8888;

try {

Socket socket = new Socket(targetHost, targetPort);

PrintWriter pw = new PrintWriter(new BufferedWriter(new OutputStreamWriter(

socket.getOutputStream())));

pw.println("stop");//发送停止命令

pw.flush();

socket.close();

} catch (UnknownHostException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

}

控制台输出的内容为：

处理业务中...  
处理业务中...  
处理业务中...  
接收到的内容是：stop  
应用程序准备停止  
应用程序停止成功

通过向指定端口发送命令来实现关闭程序，实现比较简单，但安全性不高，如果被别人知道了这个端口以及关闭命令，就可以随时关闭正在运行的程序了。