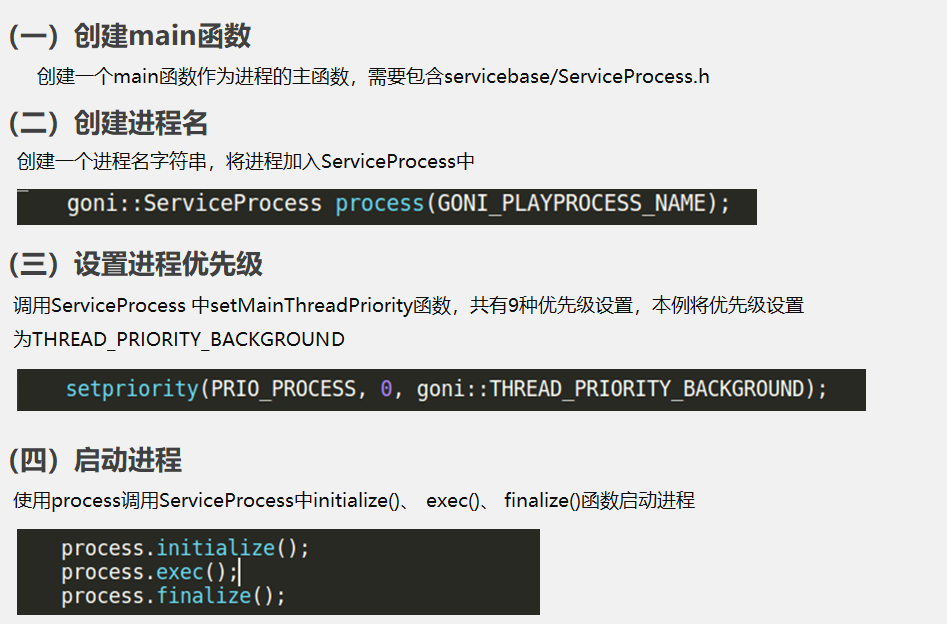
## 一 创建进程

### 1 goni平台中创建进程



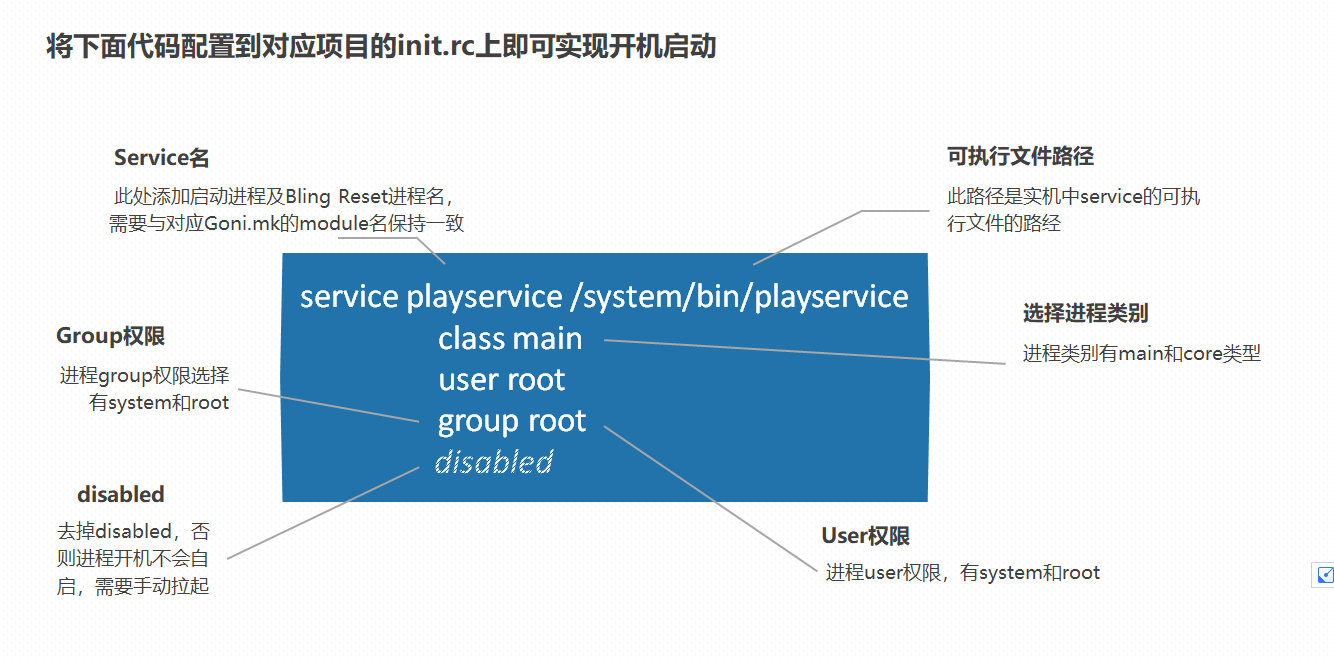
实例代码：

|  |
| --- |
| #include "servicebase/ServiceProcess.h"  #include "WeblinkService.h"  #include <sys/resource.h>  #include "utils/NCTypesDefine.h"  #include "servicebase/ServiceBaseDefines.h"  using namespace goni;  using namespace android;  int main(int argc, char \*argv[])  {  UNUSED(argc);  UNUSED(argv);  goni::ServiceProcess process("service");  // set main thread priority  setpriority(PRIO\_PROCESS, 0, THREAD\_PRIORITY\_BACKGROUND);  WeblinkService \*weblinkService = new WeblinkService();  process.addService(weblinkService);  process.initialize();  process.exec();  process.finalize();  return 0;  } |

### 2 将新的cpp文件参与整体编译



### 3 在项目中的init.rc文件中配置对应的服务。



## 二 创建service

### 1 创建service



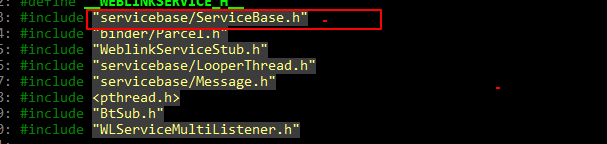
实例代码：

|  |
| --- |
| #include "servicebase/ServiceProcess.h"  #include "WeblinkService.h"  #include <sys/resource.h>  #include "utils/NCTypesDefine.h"  #include "servicebase/ServiceBaseDefines.h"  using namespace goni;  using namespace android;  int main(int argc, char \*argv[])  {  UNUSED(argc);  UNUSED(argv);  goni::ServiceProcess process("service");  // set main thread priority  setpriority(PRIO\_PROCESS, 0, THREAD\_PRIORITY\_BACKGROUND);  WeblinkService \*weblinkService = new WeblinkService();  process.addService(weblinkService);  process.initialize();  process.exec();  process.finalize();  return 0;  } |

### 2触发onStart/onStop方法

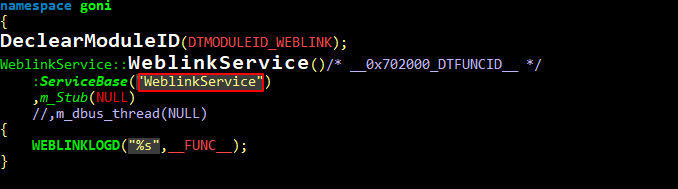
1. 实现WebLinkService

在文件中include service/serviceBase.h文件



2）**继承servicebase类**

XXXservice继承servicebase类，将service名添加servicebase中。XXXservice注册到servicemanager后，就会触发XXXservice的onstart/onstop。



**3） 创建onstart/onstop函数**

在XXXservice类中创建onstart/onstop函数，函数末尾添加onStart、 onStop结束反馈ServiceBase::onStart();/ServiceBase::onStop()。即可实现开机触发onstart、关机触发onstop。

在weblink中重写了serviceBase中的onStart/onStop方法

|  |
| --- |
| void WeblinkService::onStart() /\* \_\_0x702002\_DTFUNCID\_\_ \*/  {  WEBLINKLOGD("%s",\_\_FUNC\_\_);  m\_Stub = new WeblinkServiceStub(this);  if(NULL != m\_Stub) {  m\_WLListen = new WLServiceMultiListener(m\_Stub);  registerStub(m\_Stub);  if (m\_WLListen != NULL){  m\_btsub = new BtSub(m\_WLListen, this);  }  }  else {  DTLOG\_PTRASSERT(m\_Stub);  return;  }  ServiceBase::onStart();  m\_Stub->initialize();  }  void WeblinkService::onStop() /\* \_\_0x702003\_DTFUNCID\_\_ \*/  {  WEBLINKLOGD("%s",\_\_FUNC\_\_);  ServiceBase::onStop();  } |

## 三 进程内事件轮转处理

### 1创建工作线程

1）**创建XXXhanderThread类**

创建一个XXXhanderThread类，包含startThread()、stopThread()、getLooper()三个函数

需要包含looper/MRunnableThread.h

|  |
| --- |
|  |

1. **编写XXXhanderThread成员函数**
2. **执行startThread()**

### 2把事件传递到工作线程

1. **创建函数A将线程发送postRunnable中**

创建一个函数，将线程放入队列中，需要包含looper/MRunnableLooper.h

m\_looper.postRunnable(XXX);



1. **利用MRunnableTask创建线程模板**

MRunnableTask(this, &&PlayStub::doUsbMount, Port),第一个参数输入类对象，第二个参数输入类方法，第三个及之后参数输入类方法中输入参数，将线程模板放入(一)中线程队列中

|  |
| --- |
| GMTaskDes des(Runnable\_Pri\_N, 0, "USBMethod\_SetSurfaceInfo");  (m\_pMainThread->looper()).postRunnable(GMTaskBind(des,this, &GMUsbServiceBase::setSurfaceInfo , id, surfaceInfo)); |

1. **调用函数A**