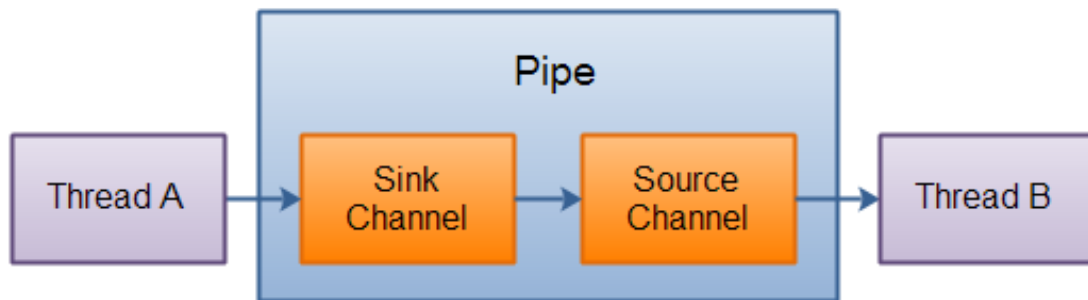


<http://wiki.jikexueyuan.com/project/java-nio-zh/java-nio-pipe.html>

一个Java NIO的管道是两个线程间单向传输数据的连接。一个管道（Pipe）有一个source channel和一个sink channel(没想到合适的中文名)。我们把数据写到sink channel中，这些数据可以同过source channel再读取出来。

下面是一个管道的示意图：



### 创建管道(Creating a Pipe)

打开一个管道通过调用Pipe.open()工厂方法，如下：

```
Pipe pipe = Pipe.open();
```

### 向管道写入数据 (Writing to a Pipe)

向管道写入数据需要访问他的sink channel:

```
Pipe.SinkChannel sinkChannel = pipe.sink();
```

接下来就是调用write()方法写入数据了：

```
String newData
    = "New String to write to
file..." + System.currentTimeMillis();
ByteBuffer buf = ByteBuffer.allocate(48);
buf.clear();
buf.put(newData.getBytes());
buf.flip();
while(buf.hasRemaining()) {
    sinkChannel.write(buf);
}
```

### 从管道读取数据 (Reading from a Pipe)

类似的从管道中读取数据需要访问他的source channel:

```
Pipe.SourceChannel sourceChannel = pipe.source();
```

接下来调用read()方法读取数据：

```
ByteBuffer buf = ByteBuffer.allocate(48);
int bytesRead = inChannel.read(buf);
```

注意这里`read()`的整型返回值代表实际读取到的字节数。