

1+1=10 简简单单，我的小屋...

个人资料



dbzhang800

访问： 1124576次
积分： 13117
等级： ?
排名： 第357名

原创： 181篇 转载： 0篇
译文： 8篇 评论： 830条

公告

本blog在IE浏览器下可能会格式错乱，请考虑非IE内核浏览器。
在多篇blog中，反斜杠被CSDN自动替换成了斜杠，请意识识别，谢谢。



本Blog中所有作品(文中额外注明版权的除外)均采用知识共享署名-非商业性使用-相同方式共享2.5进行许可。

文章搜索

Markdown博文大赛清新开启 天天爱答题 一大波C币袭来 寻找Java大牛! 大数据完美组合Spark+Scala，这样学才够值!

从QProcess说来(一)

分类： Qt 2011-10-15 17:04 16571人阅读 评论(4) 收藏 举报

windows output linux path qt null

目录(?) [+]

QProcess类的作用是启动一个外部的程序并与之交互。我们不妨看看：一个普通的控制台程序，与外界是如何交互的

简单控制台程序

一个程序如何从外界获取信息并向外输出信息呢？

方向	途径	
获取	命令行参数	int main(int argc, char **argv)
	标准输入	scanf()/getc()/getchar()/...
	环境变量	getenv()
输出	标准输出	printf() ...
	标准出错	fprintf(stderr,...)
	返回值	return(x)/exit(X)/_Exit(X)

命令行参数

大多控制台程序都接受命令行参数，一个例子：

```
#include <stdio.h>

int main(int argc, char **argv)
{
    for (int i=1; i<argc; ++i)
        printf("%s\n", argv[i]);

    return 0;
```

文章分类

C/C++ (53)

PyQt4/PySide (13)

Python (22)

Qt (122)

QTBUG (12)

tools (33)

Python源码学习 (11)

C++0x (5)

Qt5 (17)

Python Issue (1)

git (5)

cmake/qmake (17)

QWidget漫谈 (13)

latex (1)

javascript (3)

Log4Qt (3)

linux (4)

C#/.NET (9)

Qt 博客链接

齐亮的博客

白建平的博客

白净的博客

吴迪的博客

我的blogspot

我的百度博客

文章存档

2013年07月 (1)

2012年11月 (1)

2012年05月 (2)

2012年04月 (1)

2012年03月 (4)

展开

最新评论

C/C++ Strict Alias 小记

FrankHB1989: strict aliasing rule在ANSI C89 3.3。C89章号小是因为ANSI和IS...

cmake 学习笔记(一)

一动不动的葱头: 好得不得了!

QString之arg使用一则(QTBUG1

天涯小小雨: 说好的最小的数字呢那为什么不是2% 而是20%

}

编译，调用程序时可指定命令行，结果：

```
$ ./process a b "c d"

a
b
c d
```

用QProcess调用外部程序时，可直接指定命令行参数

```
QProcess process;

process.start("./process", QStringList()<<"a"<<"b");

process.start("./process a b");
```

后一种写法看起来写起来比较简洁，但是程序路径或参数中包括空格时，就不如第一种方便了。

注：在Windows平台的某些情况（比如QTBUG7620）下，你可能需要使用

- QProcess::setNativeArguments()

标准输出

对于控制台程序来说，这个可能是用的最多的了。比如上一个程序，我们遍历命令行参数，然后输到标准输出。

在控制台下，我们通常会使用重定向功能，比如：

```
$ ./process a b "c d" > out.txt
```

在QProcess下，我们使用

QProcess::readAllStandardOutput ()	获取标准输出
QProcess::setStandardOutputFile()	设置输出到的文件，相当于前面的重定向

```
QProcess process;

process.start("./process", QStringList()<<"a"<<"b");

process.readAllStandardOutput();
```

可以使用：

- QProcess::setStandardOutputProcess()

将标准输出作为另个进程的标准输入。形成 ls -l | more 这样的管道操作

由于QProcess是QIODevice的派生类，故：

- read()

从QProcess说开来(二)
baidu_27059313: 您好 我写了一个简单的c++程序A.cpp，能够成功输出“helloworld”，现在我想在B.py...

Template+=fakelib 小记
九万9w: @zhgn2:我也沮丧

Template+=fakelib 小记
九万9w: Qt 中可不止这么一个潜规则

Template+=fakelib 小记
九万9w: @surfsky:感到沮丧吧，还好有源码可以查阅各种原因+原理。也是最好的example。一个qt多...

Qt 智能指针学习
lys211: 我还需要消化啊，谢谢

从 Qt 的 delete 说开来
志愿做个PM: 受教了！

KDUpdater 入门 (Qt5)
muggle222: 我怎么找不到在哪里下载？

阅读排行

- cmake 学习笔记(一) (45746)
- Qt 线程基础(QThread、Q (28521)
- QTextCodec中的setCod (21570)
- 从 相对路径 说开来(从C+ (19439)
- 从 Qt 的 delete 说开来 (18627)
- qDebug 学习小结 (16631)
- 从QProcess说开来(一) (16561)
- Qt 智能指针学习 (15983)
- QString 乱谈(3)-Qt5与中 (15575)
- 浅谈 qmake 之 pro、pri、 (15437)

评论排行

- 用ISO C++实现自己的信 (57)
- 从 Qt 的 delete 说开来 (45)
- cmake 学习笔记(一) (38)
- 1+1=2的 blog 文章索引 (37)
- 从 相对路径 说开来(从C+ (32)
- QTextCodec中的setCod (27)
- QString乱谈(2) (20)
- Qt编码风格 (19)
- QMainWindow上下文菜单 (18)
- Qt国际化（源码含中文时 (18)

- readLine()
- ...

都可以直接用获取被调用程序的标准输出。

禁止缓冲

注意：如果写的控制台程序准备用于这种途径，且需要实时被读取标准输出，那么一般该程序内需要禁用缓冲

```
int main()  
{  
    setvbuf(stdout, (char *)NULL, _IONBF, 0);  
    return 0;  
}
```

因为在控制台运行时，标准输出是行缓的。但是使用管道或重定向以后，一般就是全缓冲。在缓冲区写满或程序退出前，你可能看不到任何输出。

标准出错

相对于标准输出，这个东西大家似乎用的比较少了。

```
#include <stdio.h>  
  
int main(int argc, char **argv)  
{  
    fprintf(stdout, "Hello STDOUT!\n");  
    fprintf(stderr, "Hello STDERR!\n");  
    return 0;  
}
```

编译运行(注意区分两个流，标准出错的文件描述符是2)：

```
$ ./process  
Hello STDOUT!  
Hello STDERR!  
$ ./process > out.txt  
Hello STDERR!  
$ ./process > out.txt 2>err.txt  
$
```

用QProcess读取标准出错，和前面标准输出是类似的：

- QProcess::readAllStandardError()
- QProcess::setStandardErrorFile()

```
QProcess process;

process.start("./process", QStringList() << "a" << "b");

process.readAllStandardOutput();
```

但是，QProcess作为QIODevice的派生类，read()/readAll()只能读标准输出，不读标准出错，有点说不过去哈。

恩QProcess在这方面足够灵活，你可以通过

- QProcess::setReadChannel()

进行选择

标准输入

这个也不用多说，使用scanf()/gfets()等函数：

```
#include <stdio.h>

int main(int argc, char **argv)
{
    char contents[1024];
    fgets(contents, 1024, stdin);
    printf("output: %s", contents);
    return 0;
}
```

运行：

```
$ ./process
1+1=2
output: 1+1=2
```

如果要输入的内容在文件内，也可以使用重定向

```
$ ./process < input.txt
output: contents of input.txt
```

在QProcess中，直接使用QIODevice的write()函数

```
QProcess process;

process.start("./process")

process.write("input");
```

也可以设置文件作为输入

关闭



- `QProcess::setStandardInputFile()`

返回值

似乎很多人分不清返回值和标准输出的概念。

```
int main()
{
    return 0;
}
```

也就是程序中 `return X/exit(X)` 等函数中指定的值了。

在Windows下，通过

```
C:\> process.exe
C:\> echo %errorlevel%
```

在Linux下，通过

```
$ ./process
$ echo $?
```

获得返回值。

在QProcess下，则通过：

- `int QProcess::execute()`
- `int QProcess::exitCode()`

获得返回值。

环境变量

程序运行时可以通过环境变量传递信息：

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    char * path = getenv("MYPATH");
    printf("MYPATH= %s\n", path);
    return 1;
}
```

运行结果(Linux):

```
$ export MYPATH="/home/dbzhang800"

$ ./process

MYPATH= /home/dbzhang800
```

Windows下:

```
C:\> set MYPATH=E:\dbzhang800

C:\> process.exe

MYPATH= E:\dbzhang800
```

在Qt下，使用 `QProcess::setProcessEnvironment()` 设置进程的环境变量

```
QProcess process;

QProcessEnvironment env;

env.insert("MYPATH", "/home/dbzhang800");

process.setProcessEnvironment(env);

process.start("./process");
```

本来只是想简单提一下这部分，然后整理**QProcess**的其他部分的。没想到这点东西竟然弄了这么长，而且还都没敢展开。不管了，先这样吧。有时间慢慢写（二、三、四）。**20111015**

- 上一篇
- Qt stylesheet 源码学习
- 下一篇
- 从QProcess说开来(二)

顶

9

踩

0

主题推荐

控制台 遍历 类 linux path

猜你在找

- Qt网络编程TCPIP二
- Tiny6410 移植QT473
- Qt学习布局管理器QLayout类
- 解决qt里点击子窗口的按钮没有反应
- ffmpeg Windows下采集摄像头一帧数据并保存为bmp图片
- 【精品课程】Spark+Scala套餐值不值，看你敢不敢学！
- 【精品课程】高安定PMP认证网络套餐5折优惠
- 【精品课程】C语言及程序设计学习套餐
- 【精品课程】学项目管理系列视频教程之基础篇(上)
- 【精品课程】Java Web高级技术

