

用 popen 函数操作其它程序的输入和输出

一、函数介绍

1.1 函数原型:

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

```
FILE *popen(const char *command, const char *open_mode);
```

1.2 说明

popen 函数允许一个程序将另一个程序作为新进程启动，并可以传递数据给它或者通过它接收数据。command 字符串是要运行的程序名和相应参数（比如：ls 或 ls -l），openmode 必须是 r 或 w。如果是 r，被调用程序的输出可以被调用它的程序使用；如果是 w，调用程序就可以用 fwrite 向被调用程序发送数据作为它在标准输入流上的输入。

二、测试程序准备

这里准备两个很简单的程序供下面测试使用。

2.1 输出测试程序

```
// outputTest.c
#include <stdio.h>

int main()
{
    printf("Just a test ! \n");
    return 0;
}
```

主要是实现向标准输出设备输出字符串，供下面的程序进行测试。

2.2 输入测试程序

```
// inputTest.c
#include <stdio.h>

int main()
{
    char buf[1024] = {0};
    scanf("%s", buf);
    printf("your input : %s\n", buf);
    return 0;
}
```

主要是实现从标准输入设备输入字符串并输出，供下面的程序进行测试。

三、popen 操作示例（C 代码）

3.1 获得程序输出

以 outputTest 程序来测试，代码如下：

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

int main()
{
    FILE *read_fp;
```

```

char buffer[BUFSIZ + 1];
int chars_read;
memset(buffer, '\0', sizeof(buffer));
read_fp = popen("./outputTest", "r");
if(read_fp != NULL)
{
    chars_read = fread(buffer, sizeof(char), BUFSIZ, read_fp);
    if(chars_read > 0)
    {
        printf("Output was : \n%s\nDone\n", buffer);
    }
    pclose(read_fp);
    exit(EXIT_SUCCESS);
}
exit(EXIT_FAILURE);
}

```

运行效果如下:

```

[root@host232 test2]# ./popen1
Output was :
Just a test !

```

Done

```

[root@host232 test2]#
E-Mail : Mike_Zhang@live.com

```

这里主要用r参数获得被调用程序的输出。

3.2 给其它程序传参数

以inputTest来测试,代码如下:

```

#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main()
{
    FILE *write_fp;
    char buffer[BUFSIZ + 1];

    sprintf(buffer, "Once...\n");
    write_fp = popen("./inputTest", "w");
    if(write_fp != NULL)
    {
        fwrite(buffer, sizeof(char), strlen(buffer), write_fp);
        pclose(write_fp);
        exit(EXIT_SUCCESS);
    }
    exit(EXIT_FAILURE);
}

```

运行效果如下:

```
[root@host232 test2]# ./popen2
your input : Once...
[root@host232 test2]#
```

E-Mail : Mike_Zhang@live.com

这里主要用 w 参数向被调用程序传参数。

四、popen 操作示例 (python 代码)

其实 python 也可以这样玩的。

4.1 获得程序输出

还以上文提到的 outputTest 程序为例，代码如下：

```
#!/usr/bin/python

import os
#var = os.popen('ls -l').read()
var = os.popen('./outputTest').read()
print var
```

运行效果如下：

```
[root@host232 test2]# ./popen1.py
Just a test !

[root@host232 test2]#
```

E-Mail : Mike_Zhang@live.com

4.2 给其它程序传参数

还以上文提到的 inputTest 程序为例，代码如下：

```
#!/usr/bin/python

import os
os.popen('./inputTest', 'w').write("test")
```

运行效果如下：

```
[root@host232 test2]# ./popen2.py
your input : test
[root@host232 test2]#
```

E-Mail : Mike_Zhang@live.com