Pipe

张海宁

贵州大学

hnzhang1@gzu.edu.cn

May 29, 2018

Overview

1. Pipe
   - What is a Pipe
2. Process pipe
3. Appendix

Pipe

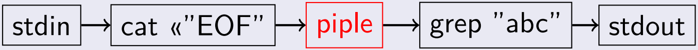
What is a Pipe

Pipe

We use the term *pipe* to mean connecting a data flow from one process to another.

shell command

```
cat <<"EOF" | grep "abc"
```



stdin → cat «"EOF" → piple → grep "abc" → stdout

Process pipe

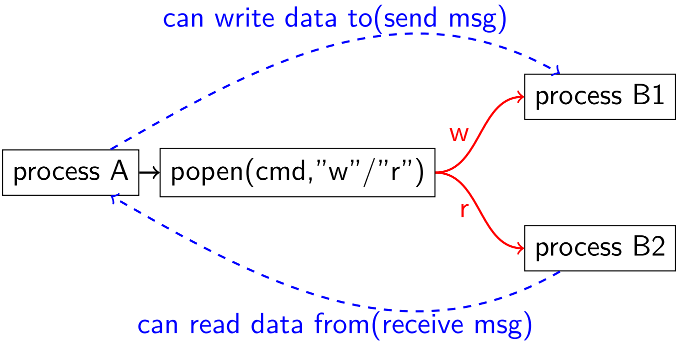
Process pipe

Perhaps the simplest way of passing data between two programs is with the *popen* and *pclose* functions.

原型

```
#include <stdio.h>
FILE *popen(const char *command, const char *open_mode);
int pclose(FILE *stream_to_close);
```

popen


can write data to(send msg)
process B1
w
process A
popen(cmd,"w"/"r")
r
process B2
can read data from(receive msg)

read data from child process

12-pipeRead.c

```
#include<unistd.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main(){
 FILE * f; char buf[1000]; int len;
 memset(buf,'\0',1000);
 f=popen("uname -a","r"); sleep(5);
 if(f==NULL){
  perror("error in create another process!"); exit(-1); }
 len = fread(buf,sizeof(char),1000,f);
 if(len>0){
  printf("the out put of uname -a is:\n%s\n",buf); }
 printf("finished!"); pclose(f); exit(0);
}
```

read data from child process

13-pipeRead.c

```
./pipe &
[2] 5254
[1]   Done                    ./pipe
$ ps -j
USER           PID  PPID  PGID   SESS JOBC STAT COMMAND
hainingzhang  1583  1582  1583      0    1 S     -bash
hainingzhang  5254  1583  5254      0    1 S     ./pipe
hainingzhang  5255  5254  5254      0    1 Z    (uname)
$ the out put of uname -a is:
Darwin HainingdeMacBook-Pro.local 17.5.0 Darwin Kernel
 Version 17.5.0: Mon Mar  5 22:24:32 PST 2018;
  root:xnu-4570.51.1~1/RELEASE_X86_64 x86_64

finished!
```

send data to child process

13-pipeWrite.c

```
#include<unistd.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main(){
 FILE * f; char buf[1000]; int len;
 memset(buf,'\0',1000);
 sprintf(buf,"I can say a, b ,c and d.");
 f=popen("wc -w","w");
 if(f==NULL){
  perror("error in create another process!");
  exit(-1); }
 fwrite(buf,sizeof(char),strlen(buf),f);
 printf("finished!"); pclose(f); exit(0);
}
```

作业

编写一个 socket 程序，要求：

1. 使用 TCP 协议实现
2. 客户端可以和服务器端进行通信
3. 当用户输入 end 时，本客户端退出结束
4. 进阶要求：
    1. 多个客户端可以同时分别和服务端通信
    2. 实现一个类似聊天室的功能

The End

Appendix

本课程相关资源下载

1. ppt
   `https://github.com/gmsft/ppt/tree/master/linux`
2. 实验指导书
   `https://github.com/gmsft/ppt/tree/master/book/linux`

about man page

The manual is generally split into eight numbered sections, organized as follows (on Research Unix, BSD, macOS and Linux):

section	description
1	General commands
2	System calls
3	Library function(C standard library)
4	Special files(devices) and drivers
5	File formats and conventions
6	Games and screensavers
7	Miscellanea
8	System administration commands and daemons

Table: man page

在终端中运行 man read 与 man 2 read，观察其输出的区别。