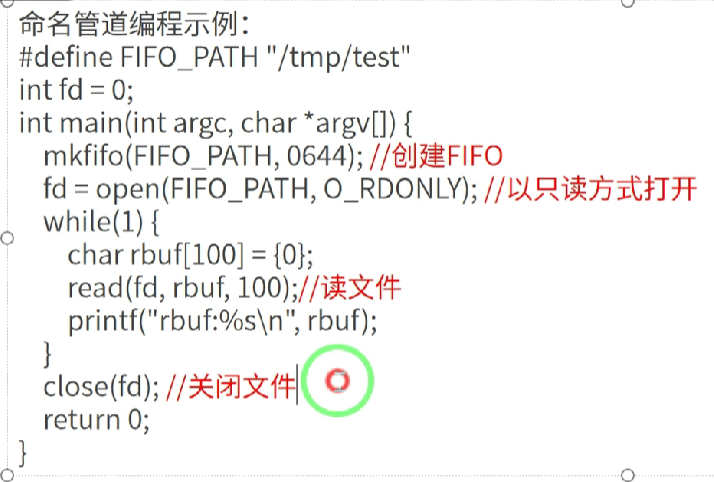


mkfifo(pathname,mode\_t mode);

命名管道使用文件IO open，write，read， close 来实现打开，写读，关闭等操作。



#define FIFO\_PATH “/tmp/test“

Int fd =0;

int main(int argc, char \* argv[]){

mkfifo(FIFO\_PATH,0644);

fd = open(FIFO\_PATH,O\_RDONLY);//以只读方式打开

while(1){

char rbuf[100] = {0};

read(fd,rbuf,100);

printf(“rbuf : %s\n”,rbuf);

}

close（fd）;

return 0;

}

#include<stdio.h>

#include<errno.h>

#include<string.h>

#include<fcntl.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<unistd.h>

#include<stdlib.h>

#define FIFO\_PATH "/home/ljaep/studio\_base/linux\_stu/work/fifo"

int fifo\_rtest(){

int fid = mkfifo(FIFO\_PATH ,0644);

if((fid)==-1 && errno!=EEXIST){

perror("mkfifo");

return -1;

}

int fd = open(FIFO\_PATH,O\_RDONLY);

if(fd==-1){

perror("open");

return -1;

}

while(1){

char buff[100] = {0};

read(fd,buff,sizeof(buff));

printf("buf:%s\n",buff);

}

return 0;

}

int fifo\_wtest(char \*wm){

int fid = mkfifo(FIFO\_PATH ,0644);

if((fid)==-1 && errno!=EEXIST){

perror("mkfifo");

return -1;

}

int fd = open(FIFO\_PATH,O\_WRONLY);

if(fd==-1){

perror("open");

return -1;

}

while(1){

write(fd,wm,strlen(wm));

sleep(1);

}

return 0;

}

int test\_pro(){

int fid = mkfifo(FIFO\_PATH ,0644);

if((fid)==-1 && errno!=EEXIST){

perror("mkfifo");

return -1;

}

int pid = fork();

if(pid ==0){

int fd = open(FIFO\_PATH,O\_RDONLY);

if(fd==-1){

perror("open");

return -1;

}

while(1){

char buff[100] = {0};

read(fd,buff,sizeof(buff));

printf("buf:%s\n",buff);

}

}

else if(pid > 0){

int fd = open(FIFO\_PATH,O\_WRONLY);

if(fd==-1){

perror("open");

return -1;

}

while(1){

char \*wm = "sun";

write(fd,wm,strlen(wm));

sleep(1);

}

}

else{

perror("fork");

}

}

int main(int argc ,char \* argv[]){

#if 1

test\_pro();

#else

int mode = atoi(argv[1]);

char \* wm = argv[2];

if(mode==0){

fifo\_rtest();

}

else{

fifo\_wtest(wm);

}

return 0 ;

#endif

return 0;

}

sy