1. 将标准输出重定向到log1文件

#include <func.h>

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

{

    ARGS\_CHECK(argc,2)

    int fd;

    fd=open(argv[1],O\_RDWR);

    ERROR\_CHECK(fd,-1,"open")

    printf("\n");//刷新标准输出

    int fd1=dup2(fd,STDOUT\_FILENO);//dup2第二参数是指要复制到什么位置

    printf("fd1=%d\n",fd1);

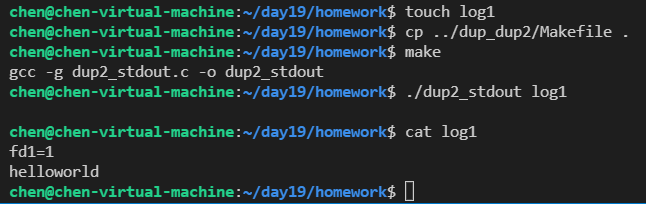
    printf("helloworld\n");

    close(fd);

    close(fd1);

    return 0;

}



1. 通过mkfifo命令新建两个管道文件1.fifo，2.fifo，然后两个进程通过管道文件实现即时聊天效果，就是双方想说就说（用上select）

#include <func.h>

//聊天1，先打开1号管道的读端，再打开2号管道的写端

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

{

    ARGS\_CHECK(argc,3)

    int fdr,fdw;

    fdr=open(argv[1],O\_RDONLY);

    ERROR\_CHECK(fdr,-1,"open1")

    fdw=open(argv[2],O\_WRONLY);

    ERROR\_CHECK(fdw,-1,"open2")

    printf("I am chat1\n");

    //select聊天

    char buf[128]={0};

    fd\_set rdset;

    int ret;

    int ready\_fdnum;

    while(1)

    {

        FD\_ZERO(&rdset);

        FD\_SET(STDIN\_FILENO,&rdset);

        FD\_SET(fdr,&rdset);

        ready\_fdnum=select(fdr+1,&rdset,NULL,NULL,NULL);//rdset是传入传出参数

        if(FD\_ISSET(STDIN\_FILENO,&rdset))//标准输入有数据

        {

            memset(buf,0,sizeof(buf));

            ret=read(STDIN\_FILENO,buf,sizeof(buf));

            if(0==ret)

            {

                printf("I want to leave\n");

                break;

            }

            write(fdw,buf,strlen(buf)-1);

        }

        if(FD\_ISSET(fdr,&rdset))

        {

            memset(buf,0,sizeof(buf));

            ret=read(fdr,buf,sizeof(buf));

            if(0==ret)

            {

                printf("byebye\n");

                break;

            }

            printf("get=%s\n",buf);

        }

    }

    close(fdr);

    close(fdw);

    return 0;

}

#include <func.h>

//聊天2，先打开1号管道的写端，再打开2号管道的读端

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

{

    ARGS\_CHECK(argc,3)

    int fdr,fdw;

    fdw=open(argv[1],O\_WRONLY);

    ERROR\_CHECK(fdw,-1,"open1")

    fdr=open(argv[2],O\_RDONLY);

    ERROR\_CHECK(fdr,-1,"open2")

    printf("I am chat2\n");

    //select聊天

    char buf[128]={0};

    fd\_set rdset;

    int ret;

    int ready\_fdnum;

    struct timeval t;

    while(1)

    {

        FD\_ZERO(&rdset);

        FD\_SET(STDIN\_FILENO,&rdset);

        FD\_SET(fdr,&rdset);

        bzero(&t,sizeof(t));

        ready\_fdnum=select(fdr+1,&rdset,NULL,NULL,NULL);//rdset是传入传出参数

        if(FD\_ISSET(STDIN\_FILENO,&rdset))//标准输入有数据

        {

            memset(buf,0,sizeof(buf));

            ret=read(STDIN\_FILENO,buf,sizeof(buf));

            if(0==ret)

            {

                printf("I want to leave\n");

                break;

            }

        write(fdw,buf,strlen(buf)-1);

        }

        if(FD\_ISSET(fdr,&rdset))

        {

            memset(buf,0,sizeof(buf));

            ret=read(fdr,buf,sizeof(buf));

            if(0==ret)

            {

                printf("byebye\n");

                break;

            }

            printf("get=%s\n",buf);

        }

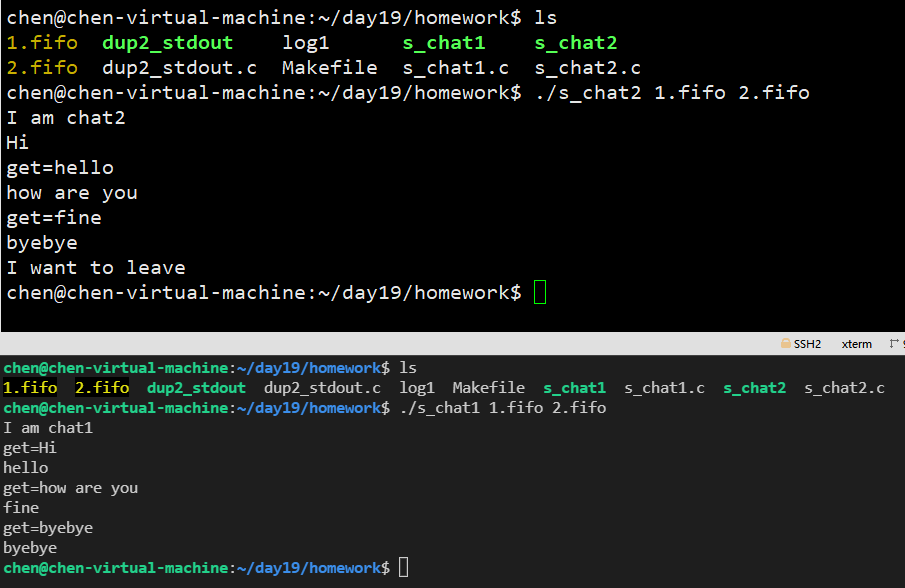
    }

    close(fdr);

    close(fdw);

    return 0;

}



1. 编写程序，打印当前进程pid，父进程pid，真实用户ID，真实组ID，有效用户ID，有效组ID

#include <func.h>

int main()

{

    printf("pid:%d  ppid:%d\n",getpid(),getppid());

    printf("uid:%d  gid:%d  euid:%d  egid:%d\n",getuid(),getgid(),geteuid(),getegid());

    return 0;

}

IMG_256

1. luke用户编写一个程序，用来往file文件中写入内容hello，然后编译后，luke给可执行文件增加s权限，这是xiognda执行该程序，能够往file文件中写入hello

#include <func.h>

int main()

{

    int fd;

    fd=open("file",O\_RDWR);

    ERROR\_CHECK(fd,-1,"open");

    write(fd,"hello",5);

    close(fd);

    return 0;

}

