管道也叫无名管道，它是是 UNIX 系统 IPC（进程间通信） 的最古老形式

# 管道的操作

所需头文件：

#include <unistd.h>

**创建管道**

int pipe(int filedes[2]);

功能：

创建无名管道。

参数：

**filedes**:

存放了管道的文件描述符 filedes[0]、filedes[1]。**其中 fd[0] 固定用于读管道，而 fd[1] 固定用于写管道，使用IO函数读写管道**

返回值：

成功：0

失败：-1

下面我们写这个一个例子，子进程通过无名管道给父进程传递一个字符串数据：

#include <stdio.h>

#include <string.h>

#include <unistd.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/wait.h>

int main(int argc, char \*argv[])

{

    int fd\_pipe[2] = {0};

    pid\_t pid;

    if( pipe(fd\_pipe) < 0 ){// 创建无名管道

        perror("pipe");

    }

    pid = fork(); // 创建进程

    if( pid < 0 ){ // 出错

        perror("fork");

        exit(-1);

    }

    if( pid == 0 ){ // 子进程

        char buf[] = "I am sunplus";

        // 往管道写端写数据

        write(fd\_pipe[1], buf, strlen(buf));

        \_exit(0);

    }

    else if( pid > 0){// 父进程

        wait(NULL); // 等待子进程结束，回收其资源

        char str[50] = {0};

        // 从管理里读数据

        read(fd\_pipe[0], str, sizeof(str));

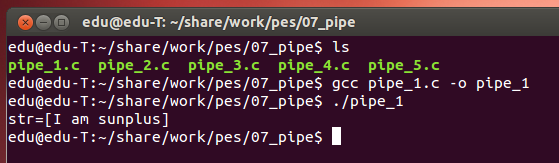
        printf("str=[%s]\n", str); // 打印数据

    }

    return 0;

}

运行结果：



# 管道的特点

当管道里没有数据，另一个进程默认用 read() 函数从管道中读数据是阻塞的。

编程时可通过 fcntl() 函数设置文件的阻塞特性。

设置为阻塞：fcntl(fd, F\_SETFL, 0);

设置为非阻塞：fcntl(fd, F\_SETFL, O\_NONBLOCK);