



北京邮电大学

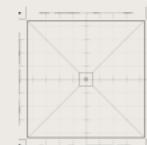
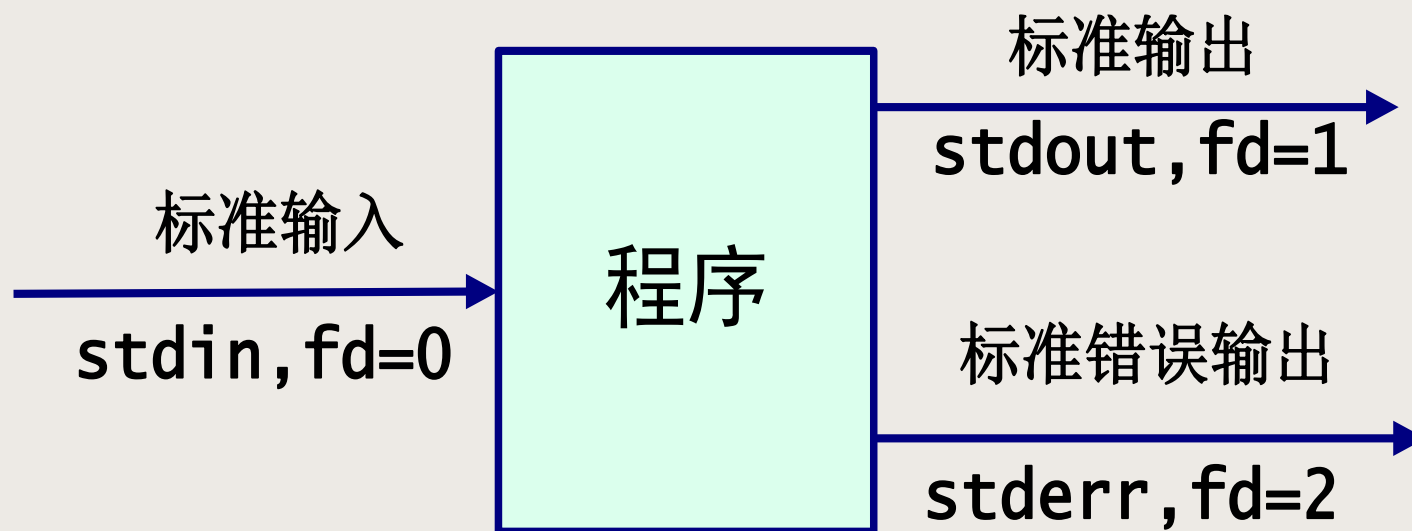
Beijing University of Posts and Telecommunications



输出重定向与管道

蒋砚军 北京邮电大学计算机学院

► 程序的标准输入/输出



► 使用系统调用（原始I/O）的程序举例



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

int main(void)          /* 使用原始I/O */
{
    static char *str1 = "string1\n";
    static char *str2 = "string2\n";
    int i;
    for (i = 0; i < 10; i++) {
        write(1, str1, strlen(str1));
        write(2, str2, strlen(str2));
    }
}
```

► 使用C语言库函数（缓冲I/O）的程序举例



```
#include <stdio.h>
/* FILE*类型的变量stdin, stdout和stderr */

int main(void) /* 使用缓冲I/O */
{
    static char *str1 = "string1\n";
    static char *str2 = "string2\n";
    int i;
    for (i = 0; i < 10; i++) {
        printf("%s", str1); /*或:fprintf(stdout, "%s", str1);*/
        fprintf(stderr, "%s", str2);
    }
}
```

► stdout输出重定向

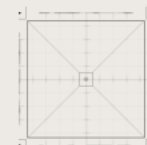


> *filename*

将stdout重定向到文件*filename*，文件已存在则先清空（覆盖方式）

>> *filename*

将stdout重定向追加到文件*filename*尾



► stderr输出重定向



2> *filename*

将文件句柄2重定向到文件*filename*

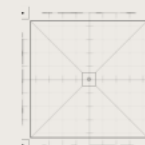
分离stdout与stderr的意义

2>&1

将文件句柄2重定向到文件描述符1指向的文件

允许对除0, 1, 2外其它文件句柄输入或输出重定向,例如:

`./myap 5< a.txt 6> b.dat`



► 输出重定向(例1)



■ `ls -l > file.list`

◆ 将命令ls标准输出stdout定向到文件file.list中

■ `cc try.c -o try 2> try.err`

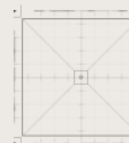
◆ 将cc命令的stderr重定向到文件try.err中

■ `try > try.out 2>try.err`

`try 1> try.out 2>try.err`

◆ 将try程序执行后的stdout和stderr分别重定向到不同的文件

■ `./stda 1> try.out 2>/dev/null`

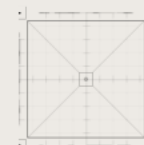
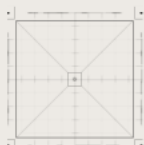
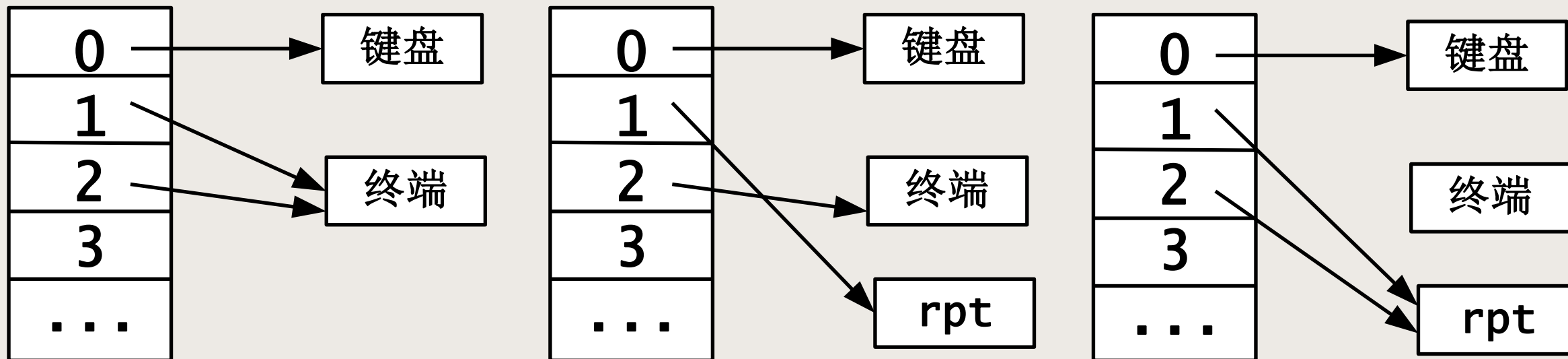


► 输出重定向(例2)



■ `./stda >rpt 2>&1`

stdout和stderr均存入文件rpt

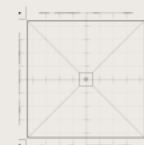
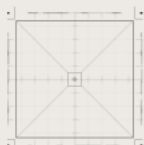
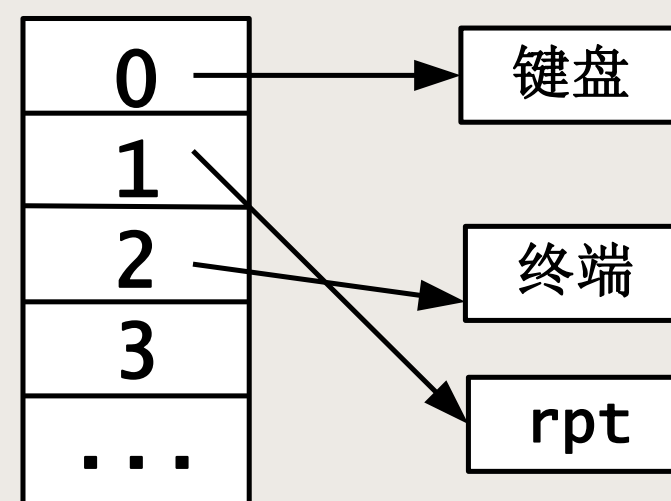
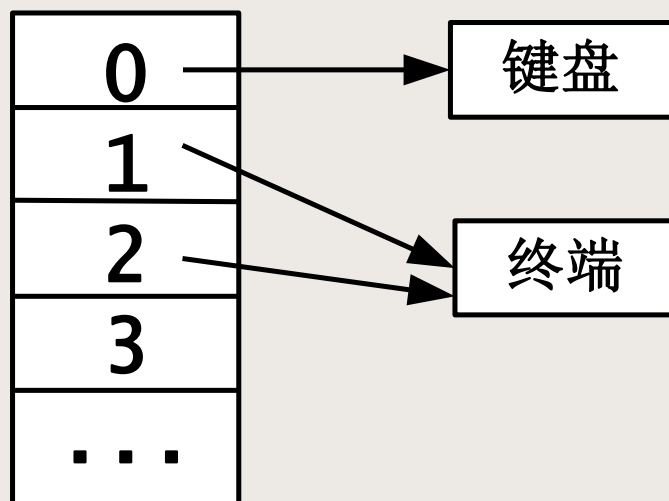
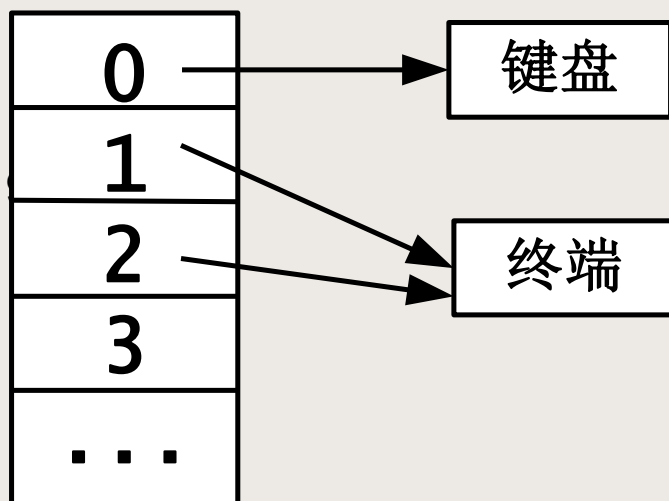


► 输出重定向(例2):错误的用法

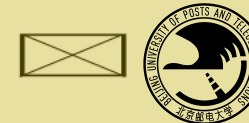


■ `./stda 2>&1 >rpt`

stderr定向到终端, stdout重定向到文件



▶ 管道

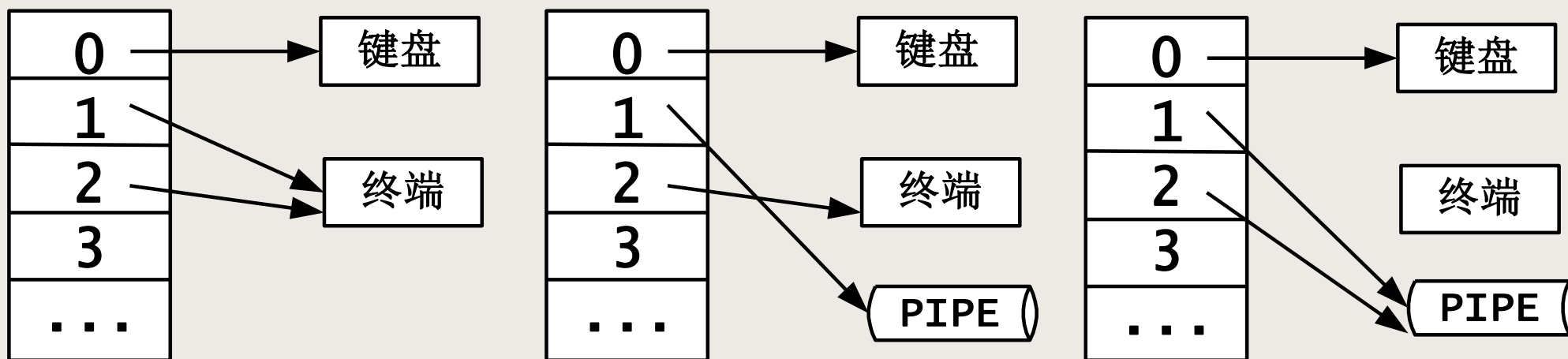


■ `ls -l | grep '^d'`

前一命令的stdout作后一命令的stdin

■ `cc try.c -o try 2>&1 | more`

前一命令的stdout+stderr作为下一命令的stdin





北京邮电大学

Beijing University of Posts and Telecommunications



谢谢