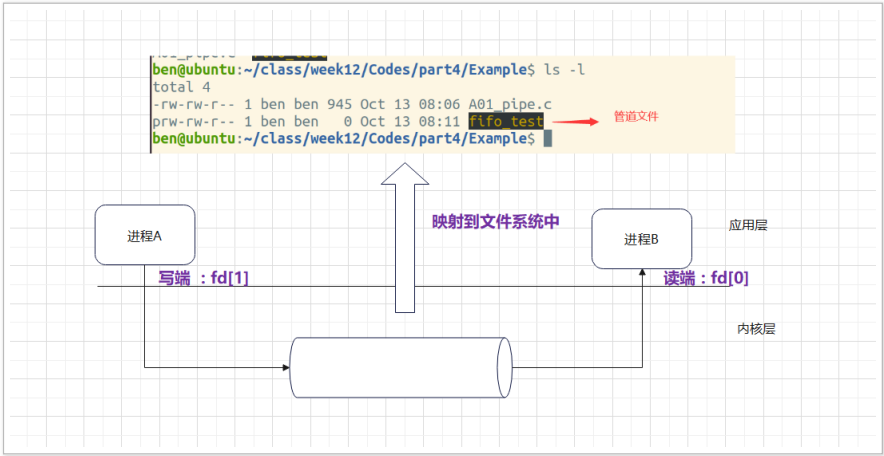


4.2 进程间通讯 - 有名管道_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 4.2 进程间通讯 – 有名管道涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

- 有名管道是在 文件系统中可见的文件, 但是不占用磁盘空间, 仍然在内存中, 可以通过 `mkfifo` 命令创建有名管道
- 有名管道与无名管道一样，在应用层是基于文件接口进行操作
- 有名管道用于 任意进程之间的通讯, 当管道为空时, 读进程会阻塞.



- 创建有名管道需要调用 `mkfifo()` 函数
 - 函数头文件
 - `#include <sys/types.h>`
 - `#include <sys/stat.h>`
 - 函数原型:
 - `int mkfifo(const char *pathname, mode_t mode);`
 - 函数参数:
 - `pathname` : 有名管道路径名
 - `mode` : 有名管道文件访问权限
 - 函数返回值:
 - 成功 : 返回 0
 - 失败 : 返回 -1, 并设置 `errno`

创建两个没有血缘关系的进程，使用有名管道进行进程间通讯

- `fifo_read.c`
 - `#include <stdio.h>`
 - `#include <stdlib.h>`
 - `#include <sys/types.h>`
 - `#include <sys/stat.h>`
 - `#include <fcntl.h>`
 - `#include <unistd.h>`
 - `#define FIFO_NAME "./fifo"`

```

int main(int argc, char *argv[])
{
    int fd;
    char rbuffer[64];

    fd = open(FIFO_NAME, O_RDWR);
    if (fd == -1) {
        perror("[ERROR] open()");
        exit(EXIT_FAILURE);
    }

    int rbytes;
    rbytes = read(fd, rbuffer, sizeof(rbuffer));
    if (rbytes > 0) {
        printf("rbytes : %d rbuffer : %s\n", rbytes, rbuffer);
    }

    close(fd);
    return 0;
}

```

- fifo_write.c

```

• #include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

#define FIFO_NAME "./fifo"

int main(int argc, char *argv[])
{
    int fd;
    char wbuffer[64] = "fifo pipe.";

    int ret;

    ret = access(FIFO_NAME, F_OK);
    if (ret == -1)
        mkfifo(FIFO_NAME, 0644);

    fd = open(FIFO_NAME, O_RDWR);
    if (fd == -1) {
        perror("[ERROR] open()");
        exit(EXIT_FAILURE);
    }

    int wbytes;
    wbytes = write(fd, wbuffer, strlen(wbuffer) + 1);
    if (wbytes < 0)
        perror("write()");

    close(fd);
    return 0;
}

```

- 注意:

- 如果有名管道的一端以只读方式打开，它会阻塞到另一端以写的方式（只写，读写）
- 如果有名管道的一端以只写方式打开，它会阻塞到另一端以读的方式（只读，读写）

- 有名管道的优缺点:

- 优点
 - 可以实现任意进程间通信，操作起来和文件操作一样
- 缺点:
 - 1. 打开的时候需要读写一起进行否则就会阻塞，管道大小是 4096 个字节
 - 2. 半双工的工作模式，如果和多个进程通信则需要创建多个管道

练习:

设计两个没有血缘关系的进程，使用有名管道一个进程获取当前系统时间给另外一个进程

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验