

OSLAB-5实验报告

OSLAB讲义

南京大学匡亚明学院 刘志刚

学号：141242022

邮箱：njuallen@foxmail.com

实验进度

总体进度：完成了JOS lab5。

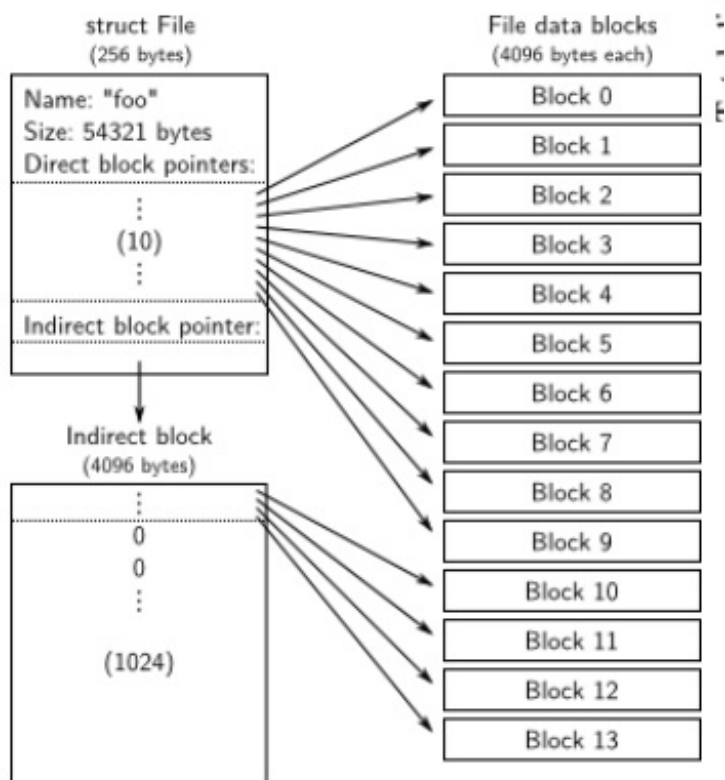
- ☑ 文件系统
- ☑ shell

对JOS文件系统的简要说明

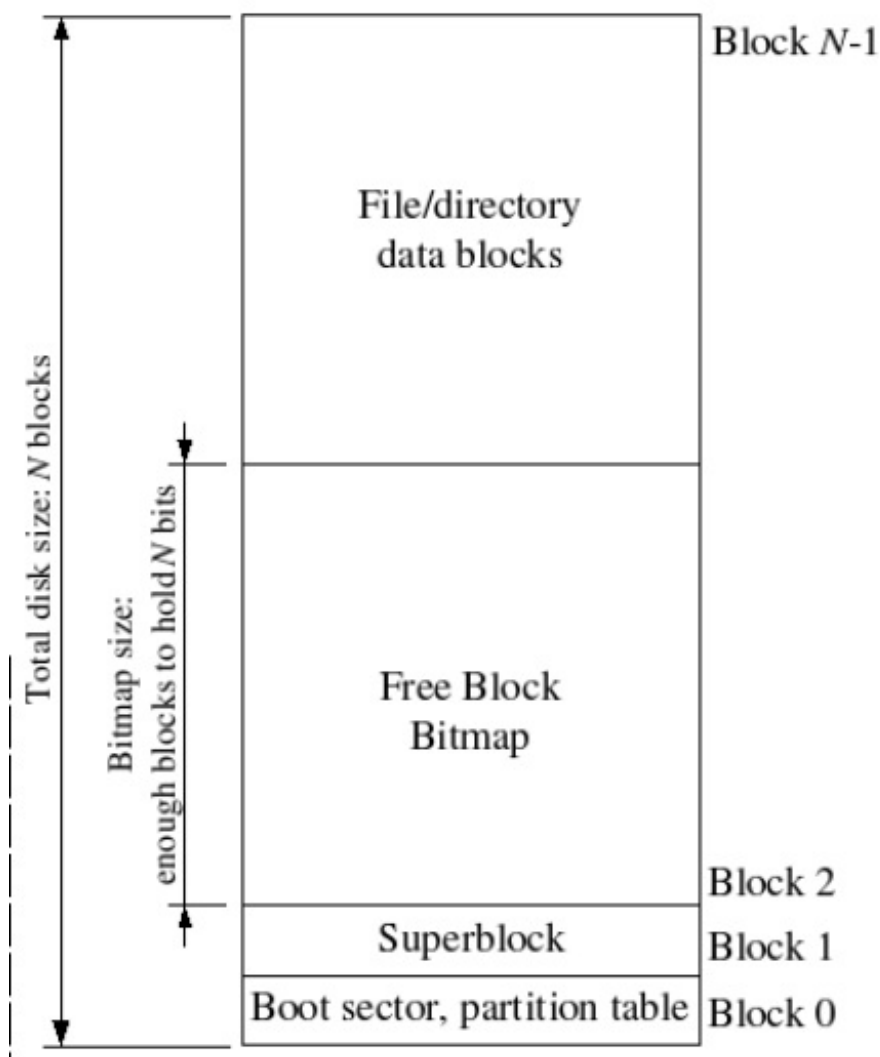
JOS采取的是微内核设计，文件系统作为一个特殊的用户程序而存在，并通过JOS之前实现的IPC机制来实现与普通用户程序的通信，并提供服务。文件系统程序之所以特殊就特殊在它是由内核在初始化时创建，并拥有读写IO端口的能力。

JOS的文件系统的格式也十分简单。包括一个super block，一个用来记录block分配与否的bitmap，并没有专门的一块inode区，而是动态地从此磁盘上取出一块，作为inode来使用。每一个文件用一个File结构体来表示，这相当于是我们常说的inode。其中包含了文件的元数据以及若干指向文件数据块的链接。其中，目录也作为文件处理，其数据块中存放的是其下的文件名，不存放其下文件的元数据。目录文件与普通文件的唯一区别是其File结构体中的type域指明了其类型为目录。

以下是file结构的简要示意图：



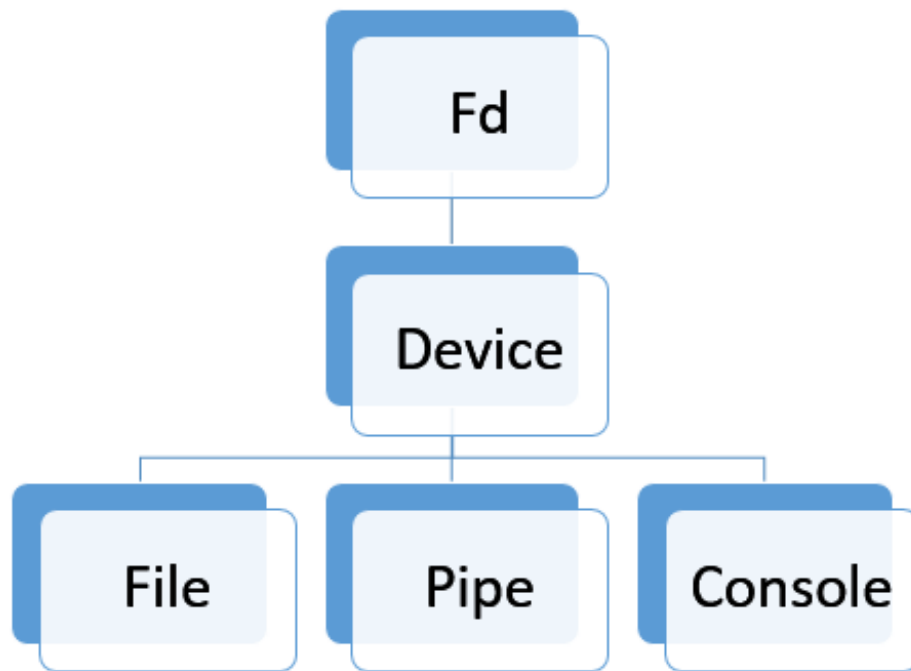
一下是文件系统在磁盘上布局的简要示意图：



JOS的文件系统在实现中充分利用了JOS之前实现的缺页错处理机制。整个磁盘被完整地映射到内存的0x10000000到0xD0000000，因此我们读写磁盘时并不担心读写磁盘哪个扇区的问题，我们直接在内存中进行操作。当发生缺页时，文件系统自己注册的缺页处理函数会帮助我们分配这一页，并把磁盘上相应内容加载进内存。

文件读写处理流程

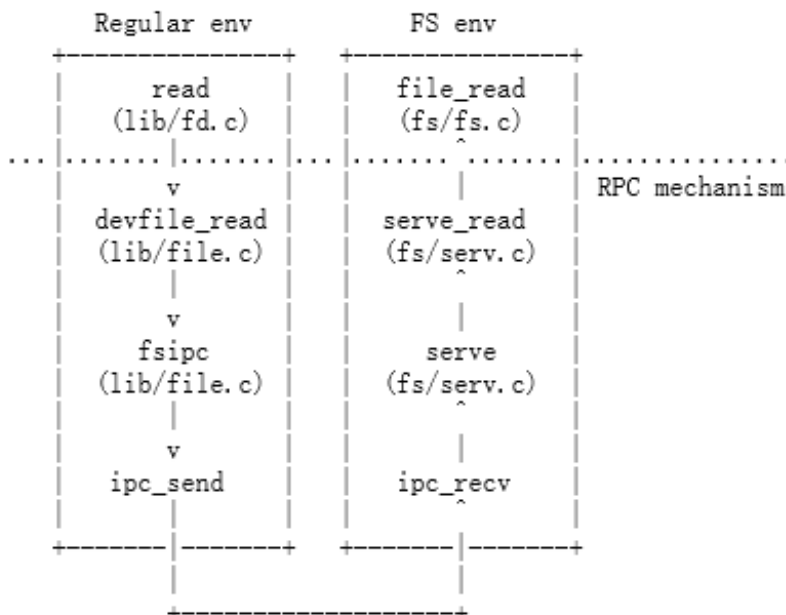
JOS中使用了一些抽象，首先是Fd文件描述符是一层抽象，我们可以对fd执行 open , close , read , seek 这一类的操作。每一个文件描述符对应的是一种设备，每一种设备都提供了一组基本的操作集合，包括读、写、seek、截断、关闭等操作。JOS中的设备文件有三种：devfile（磁盘文件），pipefile（管道文件），consfile（终端文件）。对fd的操作根据fd的类型被分发到各个设备提供的接口上处理。



以下简要介绍三种文件的处理流程：

磁盘文件：

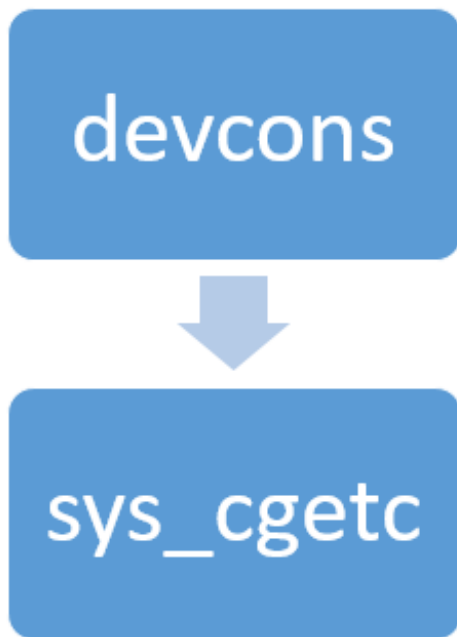
通过IPC发送指令以及可选的用来存放数据的页。fs程序完成操作并响应。



终端文件：

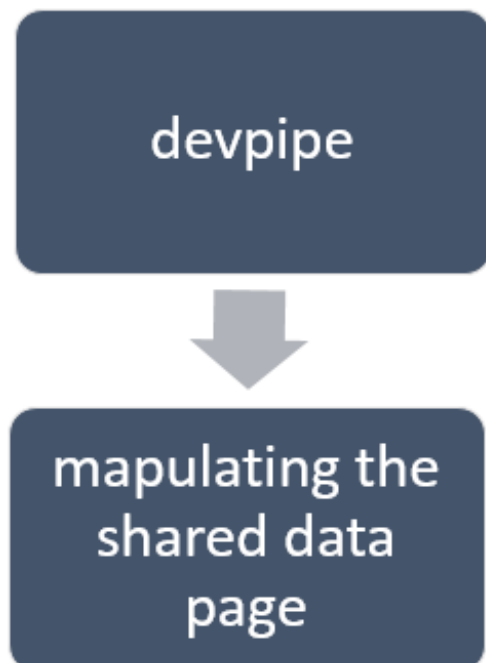
终端文件主要是用来处理面向终端的输入输出，它们最终分别是调用sys_cgetc和sys_cputc来完成。sys_cgetc从系统内部的buffer中读取字符，并读取到用户态的buffer里面做缓冲处理。

系统内部的buffer存放的是从串口和键盘输入的内容。



管道文件：

管道文件实质上是在不同进程之间共享了页做数据传递。因而对管道文件的读写最后被转化为对内存的读写。



shell

JOS中框架代码中已经实现了基于上述几类文件的读写操作，包括面向终端的输入输出，open、close、read、write，甚至于连dup都已经实现好了。结合它让我们填空的，用于创建进程的spawn（即fork与execve的结合体），我们有了充足的基础设施可以实现一个shell。

JOS中已经基于上述函数实现了一个shell，支持程序的运行，以及管道，我们只要按照要求给他加上输入输出重定向，就算完成了一个基本的shell。

结果展示

输入 `make run-icode` 即可编译并运行我们的shell。

```
Welcome to the JOS kernel, now with a file system!

icode: close /motd
icode: spawn /init
icode: exiting
init: running
init: data seems okay
init: bss seems okay
init: args: 'init' 'initarg1' 'initarg2'
init: running sh
init: starting sh
$
```

输入 `cat lorem | lorem` 我们可以尝试使用管道，并输出到标准输出。

```
$ cat lorem | cat
Lorem ipsum dolor sit amet, consectetur
adipisicing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit
in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint
occaecat cupidatat non proident, sunt in
culpa qui officia deserunt mollit anim
id est laborum.
$
```

输入 `cat <lorem >tmp` 我们可以将lorem的内容输出到tmp文件中，并以此来展示输入输出重定向的效果。

```
$ cat <lorem >tmp
$ cat tmp
Lorem ipsum dolor sit amet, consectetur
adipisicing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit
in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint
occaecat cupidatat non proident, sunt in
culpa qui officia deserunt mollit anim
id est laborum.
$
```

输入 `gobang` 即可运行我的五子棋游戏。