

# LAB5:File Systems and Spawn

0011 0010 1010 1101 0001 0100 10 OS LAB

2008-4-30

12  
45

# 主要内容

0011 0010 1010 1101 0001 0100 1011

- Lab5实习目的
- Lab5时间安排
- Lab5实习要求
- Lab5背景知识
- Lab5内容简介



# Lab5实习目的

0011 0010 1010 1101 0001 0100 1011

- 实现一个简单的**磁盘文件系统**
- 通过IPC请求来实现文件系统的访问
- 进一步加深对操作系统文件系统的理解和认识
- 实现**Spawn函数**从磁盘文件系统装入并运行一个可执行文件

45

# Lab5时间安排

0011 0010 1010 1101 0001 0100 1011

- Lab5时间： **4月30日至5月13日（2周）**
- 第1周
  - PartA FS服务进程 & PartB FS访问
- 第2周
  - Challenge、代码调试和文档

12  
45

# Lab5实习要求

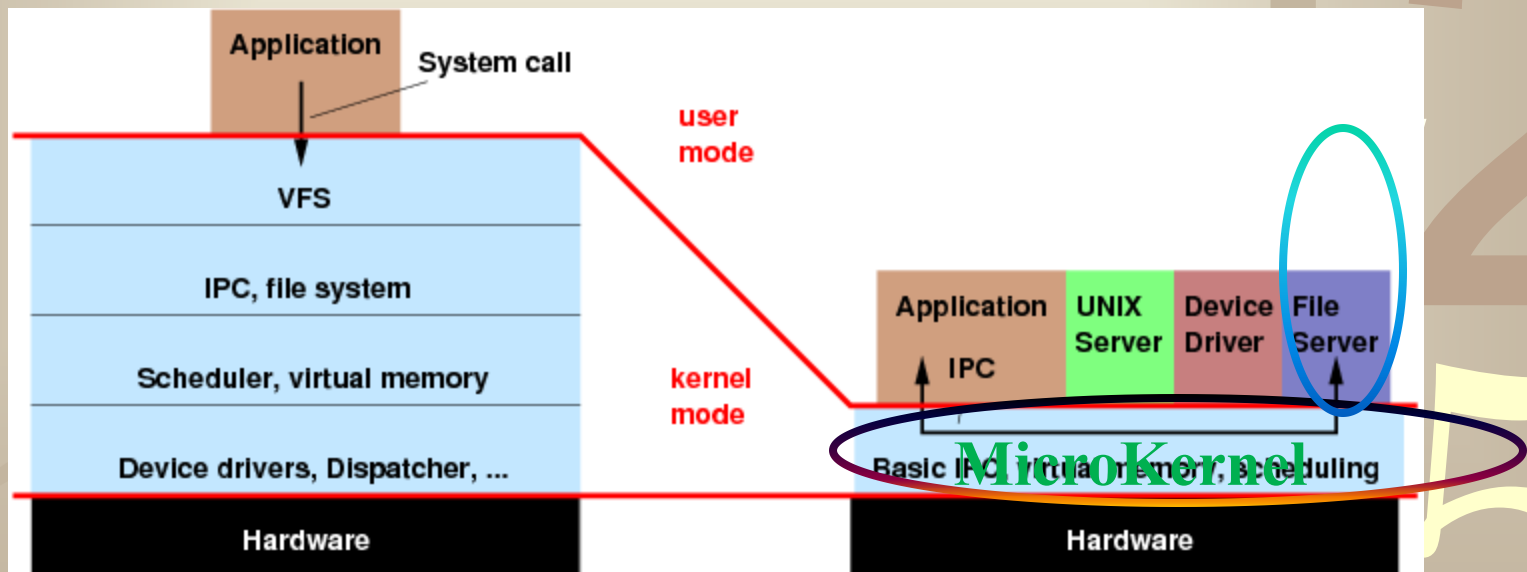
0011 0010 1010 1101 0001 0100 1011

- Exercise1—11
  - 必做，写入文档
- Questions
  - 必做，写入文档
- Challenge**2&3&6**
  - 必做(3)
- Other Challenges
  - 选作(4)

12  
45

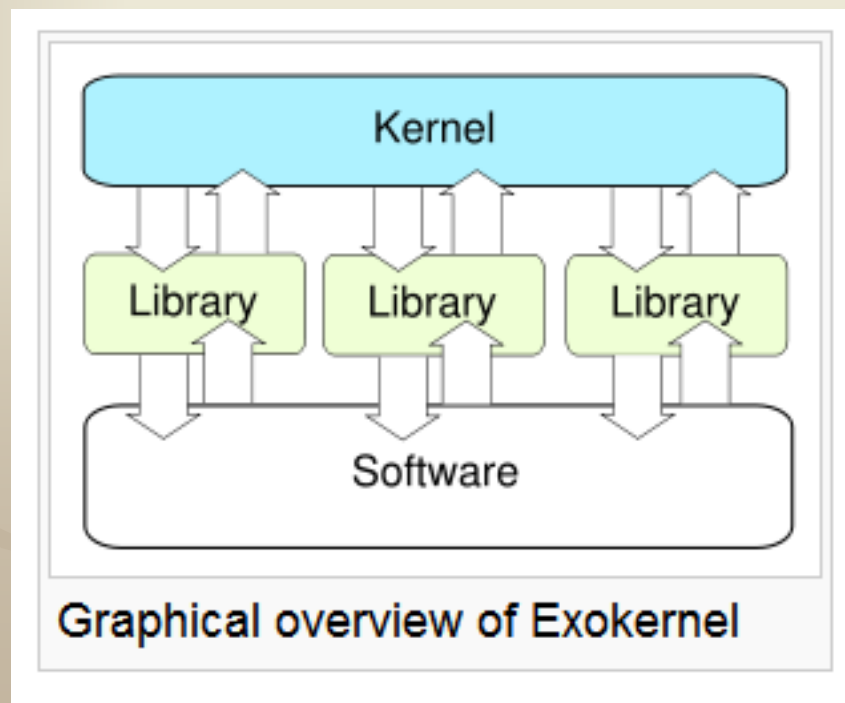
# Lab5背景知识

- Lab5中JOS的文件系统是用一种**微内核**方式（microkernel）实现的
- 文件系统（FS）本身作为一个用户进程



# Lab5背景知识

- Exec功能是通过典型的exokernel方式实现的
  - 用户空间函数库的一部分



# Lab5背景知识

0011 0010 1010 1101 0001 0100 1011

- 磁盘文件系统

- UNIX类操作系统将磁盘分成两个区域:inode区域和data区域
- inode用来保存文件的状态属性, 以及所指向数据块的指针
- data区域中包含了data块, 这里存放文件的内容和目录的元信息 (包含的文件名以及指向文件i节点的指针)



# Lab5背景知识

0011 0010 1010 1101 0001 0100 1011

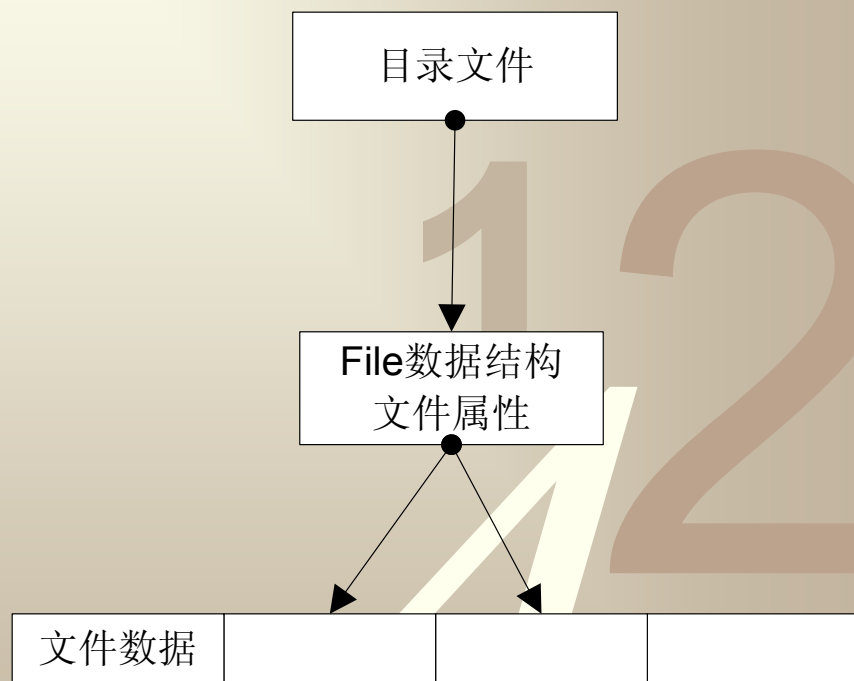
- 我们在Lab5要实现的是一个简单而强大的文件系统
  - 层次目录结构
  - create, read, write和delete操作
- J0S目前还是单用户的系统
  - 文件系统目前不支持权限和属主等属性
  - 不支持硬链接, 符号链接, 时间戳和设备文件

# Lab5背景知识

0011 0010 1010 1101 0001 0100 1011

- On-Disk文件系统结构

- 扇区 (Sector) 和块 (Block)
- 超级块 (Super block)
- 块位图 (Block bitmap)
- 文件元数据 (file meta-data)



# Lab5背景知识

0011 0010 1010 1101 0001 0100 1011

- **Sector和Block**

- Sector是磁盘执行读写操作的单位，一般是512字节
- Block是文件系统分配和使用磁盘空间的单位，是Sector的整数倍（在lab5里是4096个字节，与页的大小相等）

45

# Lab5背景知识

0011 0010 1010 1101 0001 0100 1011

- **超级块** (Super block)
  - 特定的物理位置 (磁盘的第一块或最后一块)
  - 包含描述文件系统属性的**元数据**: block的大小、磁盘大小、根目录位置、文件系统挂载的时间、上次进行磁盘检查的时间等
  - inc/fs.h中的**Super**结构定义了磁盘布局, block1为超级块, block0放boot loader和磁盘分区表

# Lab5背景知识

0011 0010 1010 1101 0001 0100 1011

- **块位图** (Block bitmap)
  - 使用位示图来**管理空闲磁盘块**，因为可以节省磁盘空间（空间换时间，带来的在内存中扫描位示图表的时间代价与之后进行的磁盘I/O相比是微不足道的）
  - **block2开始为块位示图**，涵盖范围包括所有的磁盘块，包括：block0, block1和自身

45

# Lab5背景知识

0011 0010 1010 1101 0001 0100 1011

- **文件元数据** (file meta-data)
  - file的元数据由inc/fs.h中定义的File结构描述，包括：文件名、大小、类型和指向文件所包含磁盘块的指针等；有些域仅仅在内存中才有意义，所以每次从磁盘读file结构到内存去时，都要把这些域给清空
  - File结构还保存了文件头10 (NDIRECT) 个块的全部内容, 大于10个块的文件要间接寻址

# Lab5内容简介——Part A

## • 文件系统服务器

### – 磁盘访问

- 允许服务器进程具有IO特权
- Exercise 1

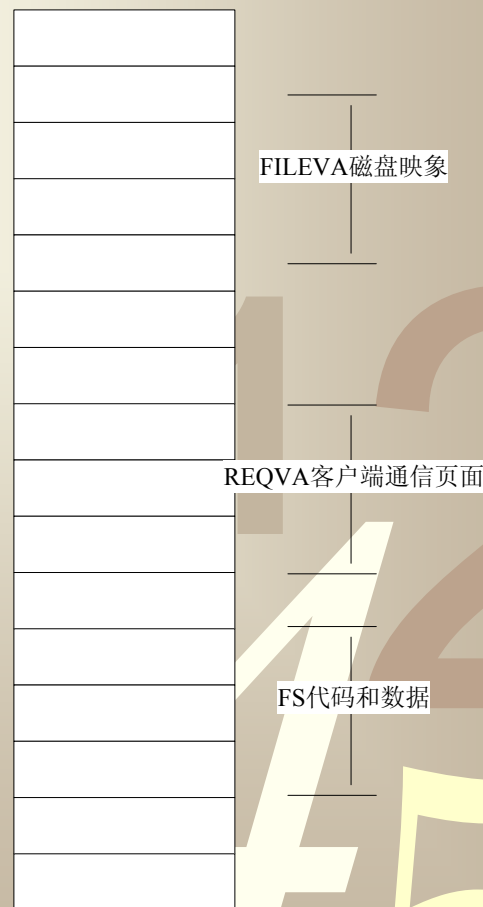
### – 磁盘块缓存

- 在内存中建立磁盘结构的映象
- Exercise 2

### – 块位图 (Exercise 3~4)

### – 文件操作 ( Exercise 5 )

### – 服务器与用户进程通信 ( Exercise 6 )



# Lab5内容简介——Part B

## • Client端的文件描述符

- 实现UNIX-类型的文件操作
- Exercise 7~9

## • Spawning Processes

- 从文件系统加载一个可执行文件到该进程中
- Exercise 10

## • Spawn arguments

- 扩展spawn使其可以传递参数到新的进程
- Exercise 11



0011 0010 1010 1101 0001 0100 1011

END

12  
45