# 文件系统数据对象创建

## 初始化

**准备工作**

1. dentry\_hashtable（）

目录项缓存，dentry\_hashtable是目录项缓存的首地址，它指向hlist\_head结构的一段内存，hlist\_head结构如下：

struct hlist\_\_head {

struct hlist\_node \*first;

}

1. inode\_hashtable

inode缓存，inode\_hashtable是inode缓存的首地址，它指向hlist\_head结构的一段内存。

1. filp\_cachep

文件对象缓存

1. mnt\_cache

mount缓存

1. mount\_hashtable

mount结构hashtable

1. mountpoint\_hashtable

挂载点hashtable

**挂载sysfs**

1. sysfs\_dir\_cachep

sysfs缓存

1. sysfs\_backing\_dev\_info

初始化sysfs\_backing\_dev\_info

1. file\_system\_type

在全局指针file\_systems指向的文件系统类型链表中找到sysfs\_fs\_type，没有找到则链接在链表的末端

1. vfsmount

创建sysfs的vfsmount

1. mount

创建sysfs的mount

1. dentry

创建sysfs的根节点目录项

1. super\_block

创建sysfs的super\_block，并将它的s\_instances通过hlist\_node结构链接到file\_system\_type的fs\_supers指向的hlist\_head数据结构

1. inode
2. kobject

为sysfs创建kobject

1. 销毁sys\_dir\_cachep

**挂载rootfs**

1. ramfs\_backing\_dev\_info
2. file\_system\_type
3. mount
4. vfsmount
5. dentry
6. super\_block
7. inode
8. mnt\_namespace

创建mnt\_namespace命名空间

1. init\_task

修改init进程的nsproxy字段的mnt\_ns为新创建的mnt\_namespace

1. 设置fs\_struct的pwd
2. 设置fs\_struct的root