

Lab14 Report

12011702 张镇涛

Q1

Block Size = 4KB

Direct Pointer: $12 \times \text{Block Size} = 48\text{KB}$

Single Indirect Pointer: $(\text{Block Size} / \text{Pointer Size}) \times \text{Block Size} = (4\text{KB} / 4\text{B}) \times 4\text{KB} = 4\text{MB}$

Double Indirect Pointer: $(\text{Block Size} / \text{Pointer Size})^2 \times \text{Block Size} = (4\text{KB} / 4\text{B})^2 \times 4\text{KB} = 4\text{GB}$

Triple Indirect Pointer: $(\text{Block Size} / \text{Pointer Size})^3 \times \text{Block Size} = (4\text{KB} / 4\text{B})^3 \times 4\text{KB} = 4\text{TB}$

因此手册第二部分中的inode最大能管理的文件大小为 **48KB+4MB+4GB+4TB**.

Q2

对于SFS inode, $\text{SFS_NDIRECT}=12$,

所以direct pointer: $12 \times \text{Block Size} = 12 \times 4\text{KB} = 48\text{KB}$.

single indirect pointer: $(\text{Block Size} / \text{Pointer Size}) \times \text{Block Size} = (4\text{KB} / 4\text{B}) \times 4\text{KB} = 4\text{MB}$

因此SFS中的inode最大能管理的文件大小为 **48KB+4MB**.

Q3

sfs_disk_inode代表了一个实际位于磁盘上的文件或目录，其记录了该文件或者目录内容存储的索引信息。

sfs_disk_entry表示一个目录inode下的某个文件项的数据结构，其存储了指向文件inode的索引值以及文件名。

对于普通文件，其sfs_disk_inode中索引值指向的 block 中保存的是文件中的数据。而对于目录，索引值指向的数据保存的是由目录下文件的文件名以及该文件对应的inode的索引值所组成的数据结构，即 sfs_disk_entry。

和 inode 相似，每个 sfs_disk_entry 也占用一个 block。