

# 电子科技大学

计算机专业类课程

## 实验报告

课程名称：计算机操作系统  
学    院：计算机科学与工程学院  
专    业：计算机科学与技术  
学生姓名：吴思扬  
学    号：2013060105016  
指导教师：薛瑞尼

日 期： 2016 年 6 月 4 日

---

# 电子科技大学 实 验 报 告 实验四

学生姓名：吴思扬

学 号：2013060105016

一、 实验项目名称： 混合索引逻辑地址到物理地址映射

- \* 条件：自定义混合索引 `inode` 结构
- \* 必须包括一次，二次，和三次间接块
- \* 逻辑块 `n` 对应物理块 `n`
- \* 输入：文件逻辑地址
- \* 输出
  1. 输出 `inode` 详细信息（间接块不展开）
  2. 物理地址（物理块号，块内偏移）

二、实验原理：

```
//  
// main.cpp  
// lab 4  
//  
//  
  
#include<iostream>  
  
using namespace std;  
void show_data(int level)  
{  
    for (int i = 0; i < level; i++)  
    {  
        for (int j = 0; j <= i; j++)
```

```

        printf(" ");
        printf("Indexes\n");
    }
    for (int j = 0; j <= level; j++)
        printf(" ");
    printf("Data\n");
}

```

```

void show_indoe()
{
    printf("indoe  "数据结构\n");
    printf("Mode\n");
    printf("Owner info\n");
    printf("Size\n");
    printf("Timestamps\n");
    printf("0-9 Direct Blocks\n");
    show_data(0);
    printf("10 Indirect Blocks\n");
    show_data(1);
    printf("11 Double Indirect\n");
    show_data(2);
    printf("12 Triple Indirect\n");
    show_data(3);
}

```

```

#define SIZEBLOCK 1024

```

```

class Cinode
{
public:
    Cinode() {}
    Cinode(const int size_block)
    {
        int num_blocks;
        int num_per_block;
        num_blocks = 0;
        num_per_block = size_block / 4;
        for (int i = 0; i < 10; i++, num_blocks++)
            Data_blocks[i] = num_blocks;
        Indirect_blocks = (int *)malloc(num_per_block * 4);
        for (int i = 0; i < num_per_block; i++, num_blocks++)
            *(Indirect_blocks + i) = num_blocks;
        num_level1 = num_blocks;
        Double_indirect = (int **)malloc(num_per_block * 4);
        for (int i = 0; i < num_per_block; i++)
        {
            *(Double_indirect + i) = (int *)mal-
loc(num_per_block * 4);
            for (int j = 0; j < num_per_block; j++,
num_blocks++)

```

```

        *((Double_indirect + i) + j) = num_blocks;
    }
    num_level2 = num_blocks;
    Triple_indirect = (int ***)malloc(num_per_block * 4);
    for (int i = 0; i < num_per_block; i++)
    {
        *(Triple_indirect + i) = (int **)mal-
loc(num_per_block * 4);
        for (int j = 0; j < num_per_block; j++)
        {
            *((Triple_indirect + i) + j) = (int *)mal-
loc(num_per_block * 4);
            for (int k = 0; k < num_per_block; k++,
num_blocks++)
            {
                *((*(Triple_indirect + i) + j) + k) =
num_blocks;
            }
        }
    }
    num_level3 = num_blocks;
}

```

```

~Cinode() {}

```

```

int Num_block(const long long int size_file, const int
size_block)
{
    int num_index;
    int num_block; //当前块数
    int num_per_block; //每个索引存放的块数
    num_index = size_file / size_block;
    num_per_block = size_block / 4;
    if (num_index < 10 && num_index >= 0)
        num_block = Data_blocks[num_index];
    else if (num_index < num_level1 && num_index >= 0)
        num_block = *(Indirect_blocks + num_index - 10);
    else if (num_index < num_level2 && num_index >= 0)
        num_block = *((Double_indirect + (num_index -
num_level1) / num_per_block) +
(num_index - num_level1) %
num_per_block);
    else if (num_index < num_level3 && num_index >= 0)
        num_block = *((*(Triple_indirect + (num_index -
num_level2) / (num_per_block * num_per_block)) + (num_index -
num_level2) % (num_per_block * num_per_block) / num_per_block)
+ (num_index - num_level2) % num_per_block);
    else
        num_block = -1;
    return num_block;
}

```

```

    int Offset(const long long int size_file, const int
size_block)
    {
        int offset;
        offset = size_file%size_block;
        return offset;
    }

void main(const int size_block)
{
    long long int size_file;
    cout << "输入文件偏移量" << endl;
    cin >> size_file;
    cout << "物理块为: " << Num_block(size_file, size_block)
<< "偏移量为: " <<
        Offset(size_file, size_block) << endl;
}

private:
    int Data_blocks[10];
    int * Indirect_blocks;
    int num_level1;
    int ** Double_indirect;
    int num_level2;
    int *** Triple_indirect;
    int num_level3;
};

int main()
{
    Cinode c(SIZEBLOCK);
    show_indoe();
    c.main(SIZEBLOCK);
}

```

## 七、实验器材（设备、元器件）:




**PC Xcode**

## 八、实验步骤及数据结果分析：

输入文件偏移量 即可计算出物理块和偏移量：

```
indoe "数据结构
Mode
Owner info
Size
Timestamps
0-9 Direct Blocks
  Data
10 Indirect Blocks
  Indexes
    Data
11 Double Indirect
  Indexes
    Indexes
      Data
12 Triple Indirect
  Indexes
    Indexes
      Indexes
        Data
输入文件偏移量
10000000
物理块为：9765偏移量为：640
Program ended with exit code: 0
```

All Output ↕

报告评分：

指导教师签字：