由于目录本身也是文件,因此它们也有大小。设我们想要计算被该树所有文件占用的磁盘区块的总数。最自然的做法是找出含于子目录/usr/mark(30)、/usr/alex(9)和/usr/bill(32)的区块的个数。于是,磁盘区块的总数就是子目录中的区块的总数(71)加上/usr使用的一个区块,共72个区块。图 4-9 中的伪码方法 size 实现这种遍历策略。

```
public int size()
{
    int totalSize = sizeOfThisFile();

    if( isDirectory())
        for each file c in this directory (for each child)
            totalSize += c.size();

    return totalSize;
}
```

图 4-9 计算一个目录大小的伪码例程

如果当前对象不是目录,那么 size 只返回它所占用的区块数。否则,被该目录占用的区块数将被加到在其所有子节点(递归地)发现的区块数中去。为了区别后序遍历策略和先序遍历策略之间的不同,图 4-10 显示每个目录或文件的大小是如何由该算法产生的。

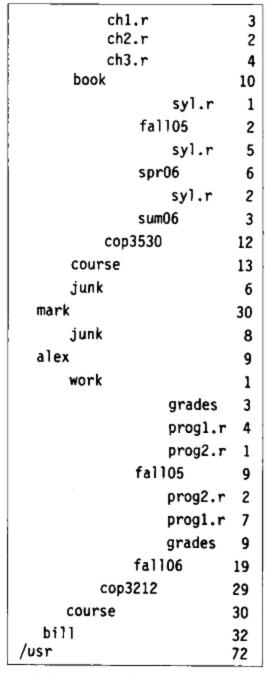


图 4-10 函数 size 的印迹