**第四部分实验 文件管理（3个学时）**

1. **实验目的**

学习MINIX1.0文件系统的实现。（对应《教程》实验十二的唯一任务）

1. **实验步骤**

*1、跟随《教程》实验十二的指导流程，阅读访问MINIX硬盘镜像文件的源代码，记录以下实验相关步骤并回答相关问题。*

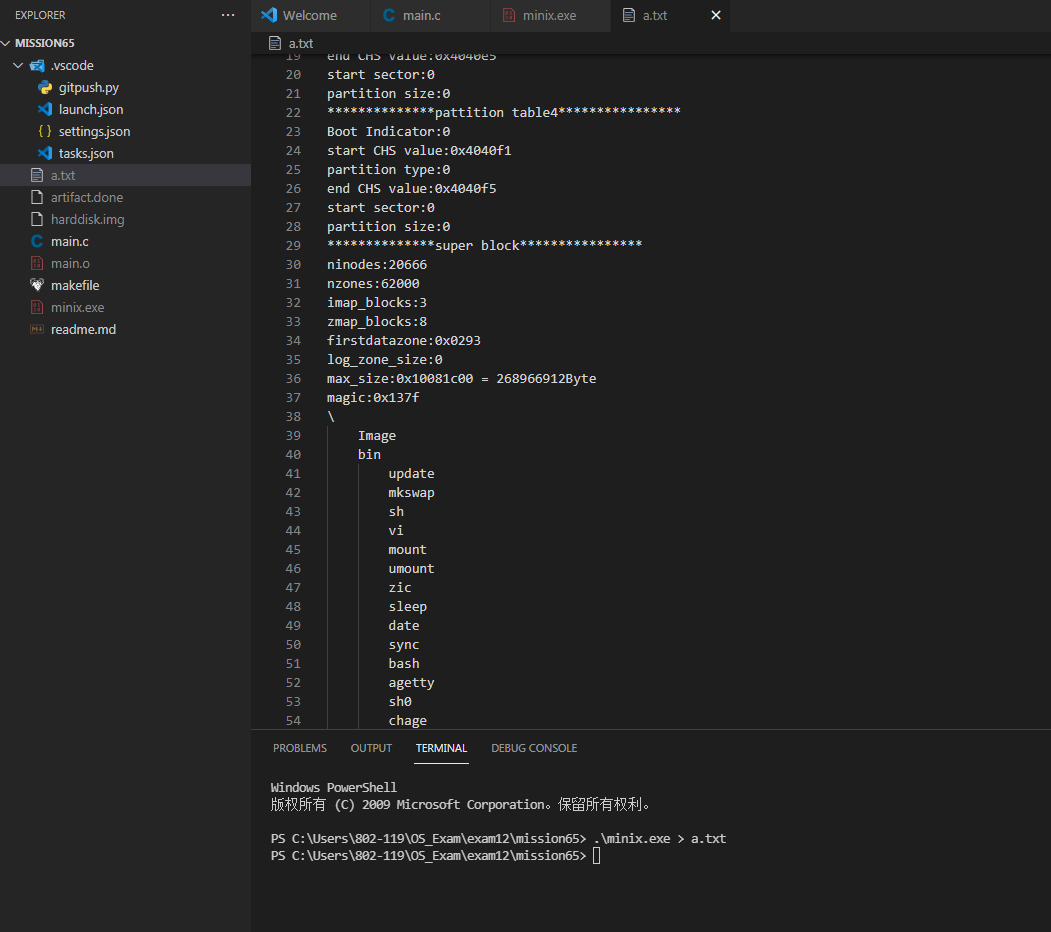
（1）分析该程序定义的主要数据对象及其字段意义？

s\_inodes short 分区中的 i 节点总数。 s\_nzones short 分区包含的逻辑块 总数。 s\_imap\_blocks short i 节点位图占用的逻辑块数。 s\_zmap\_blocks short 逻辑块位图占用的逻辑块数。 s\_firstdatazone short 数据区占用的第 一个逻辑块的块号。 s\_log\_zine\_size short log2(逻辑块包含的物理块数量)。 MINIX 1.0 文件 系统的逻辑块包含一个物理块，所以其值为 0。 s\_max\_size long 文件最大长度，以字节为单位。 s\_magic short 文件系统魔数，用以指明 文件系统的类型。MINIX 1.0 文件系统的魔数是 0x137f。i\_mode short 文件的 类型和属性(rwx 位) i\_uid short 文件宿主的用户 id i\_size long 文件长度 (以字节为单位) i\_mtime long 文件的修改时间(从 1970 年 1 月 1 日 0 时 起，以 秒为单位) i\_gid char 文件宿主的 id i\_nlinks char 链接数(有多少 个文件目录项指向该 i 节点) i\_zone[9] short 文件所占用的数据区中的逻辑 块号数组。其中 zone[0]-zone[6]是直接块号；zone[7]是一次间 接块号；zone[8] 是 二次(双重)间接块号

1. 列出该程序所实现的函数并描述其功能。

get\_physical\_block 函数的作用是将一个物理块的内容读取到缓冲区中。 get\_partition\_logical\_block 函数的作用是将第一个分区中的一个逻辑块的 内容读取到缓冲区中。 load\_inode\_bitmap 函数将硬盘镜像文件中的整个 i 节点位图都读入到了内存 中。 is\_inode\_valid 函数根据 i 节点位图中的内容判断一个 i 节点是否有效。 print\_inode 函数递归打印目录树。

1. 画出打印目录树函数的程序流程图。截图目录树输出结果。



1. **实验拓展与思考**

P142 实现更多功能中的问题，学号尾号为0和1的做第2道，学号尾号为2和3的做第3道，学号尾号为4和5的做第5道，学号尾号为6和7的做第6道，学号尾号为8和9的做第7道，并描述主要实现代码，截图输出结果。