

操作步骤：

1. Linux 环境配置：安装 Vmware Workstation 和 Ubuntu, Deepin;
2. 了解 Linux：周四上课时发现环境配置还有问题，课后根据 PPT 自行模仿操作；
3. 将第二次实验代码稍作修改，在 vim 编辑器中手动输入，适应输入基本操作；
4. 编写 makefile 文件并运行，模仿 Test 中提供的 makefile 进行操作，
间接帮助我理解了在 VS 环境下“知其然而不知其所以然”的多文件编译原理。

遇到的问题：

1. Ubuntu 安装后卡在加载界面进不去：移除虚拟机并重新创建；
2. vim 编辑器在命令模式和插入模式的切换：百度：ESC 和 i；
3. vim 编辑器光标移动指令、剪切和粘贴、翻页键的使用：百度从 CSCN 了解
\$, w/[spacebar], b/[backspace], e, O, [return]; d, y, P, p, [ctrl-d/u/f/b]用法；
4. 安装 gdb 时“无法打开锁文件，权限不够”：百度：
sudo passwd root 修改 root 密码，su 登录获得操作权限；
5. makefile 编写问题（makefile 的文件类型与如何创建）：百度：
得知 makefile 不需要后缀名，同样采用 vim 创建，
MakeFile 或 makefile 为默认名称，此外还可通过 make -f 使用其他名字；
6. freopen_s 报错问题：寻找 QQ 群里其他同学的类似提问和相关解答：
改为使用 freopen 函数并删去第一个参数&stream。

代码实现：

1. 非常淳朴的链表创建、插入、删除、排序和递归操作：
边界条件的判断和不合法操作（如空链表，长度不够）时提示信息的输出；
链表排序：两层循环嵌套、依次遍历的冒泡排序法；
两个 helper function 分别用于 Node 和 TreeNode：
void reverse_print_helper(Node* list_head)（递归无法输出换行符）
在函数 reverse_print_list 中被调用；
bool is_end(Tree_Node* T)（用于判断是否为叶节点）
在函数 insert 与 postorder 中被调用；
2. 因为是以前写过、debug 过的代码，断点调试过程中没有太多问题。
然而，发现第二周的删除节点操作没有释放/归还被删除节点的空间；
虽然运行时没有报错，但还是值得警惕，对代码中这样的问题进行了修改；
3. 发现第二周的代码中有不必要的#include 和 using namespace std。
多文件编译方面之前一直不能完全理解，通过 makefile 的编写有了更清楚的认识。