**武汉大学计算机学院**

**本科生实验报告**

**主存空间的分配与回收**

专 业 名 称 ：计算机科学与技术

课 程 名 称 ：操作系统课程设计

指 导 教 师 ：宋伟

学 生 学 号 ：2019302070035

学 生 姓 名 ：刘涛榕

二○二一年七月

**郑 重 声 明**

本人呈交的实验报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本实验报告不包含他人享有著作权的内容。对本实验报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本实验报告的知识产权归属于培养单位。

本人签名： 刘涛榕 日期：2021.7.6

**目 录**

[主存空间的分配与回收 4](#_Toc77961443)

[一、实习内容及上机实验所用平台 4](#_Toc77961444)

[**二、数据结构及代码段分析** 4](#_Toc77961445)

[**三、调试过程** 9](#_Toc77961446)

[**四、实验总结** 11](#_Toc77961447)

**主存空间的分配与回收**

**一、实习内容及上机实验所用平台**

可变分区管理方式下采用首次适应算法实现主存分配和回收

1. 可变分区方式是按作业需要的主存空间大小来分割分区的。当要装入一个作业时，根据作业需要的主存容量查看是否有足够的空闲空间，若有，则按需分配，否则，作业无法装入。假定内存大小为128K，空闲区说明表格式为：

·分区号——表示是第几个空闲分区；

·起始地址——指出空闲区的起始地址；

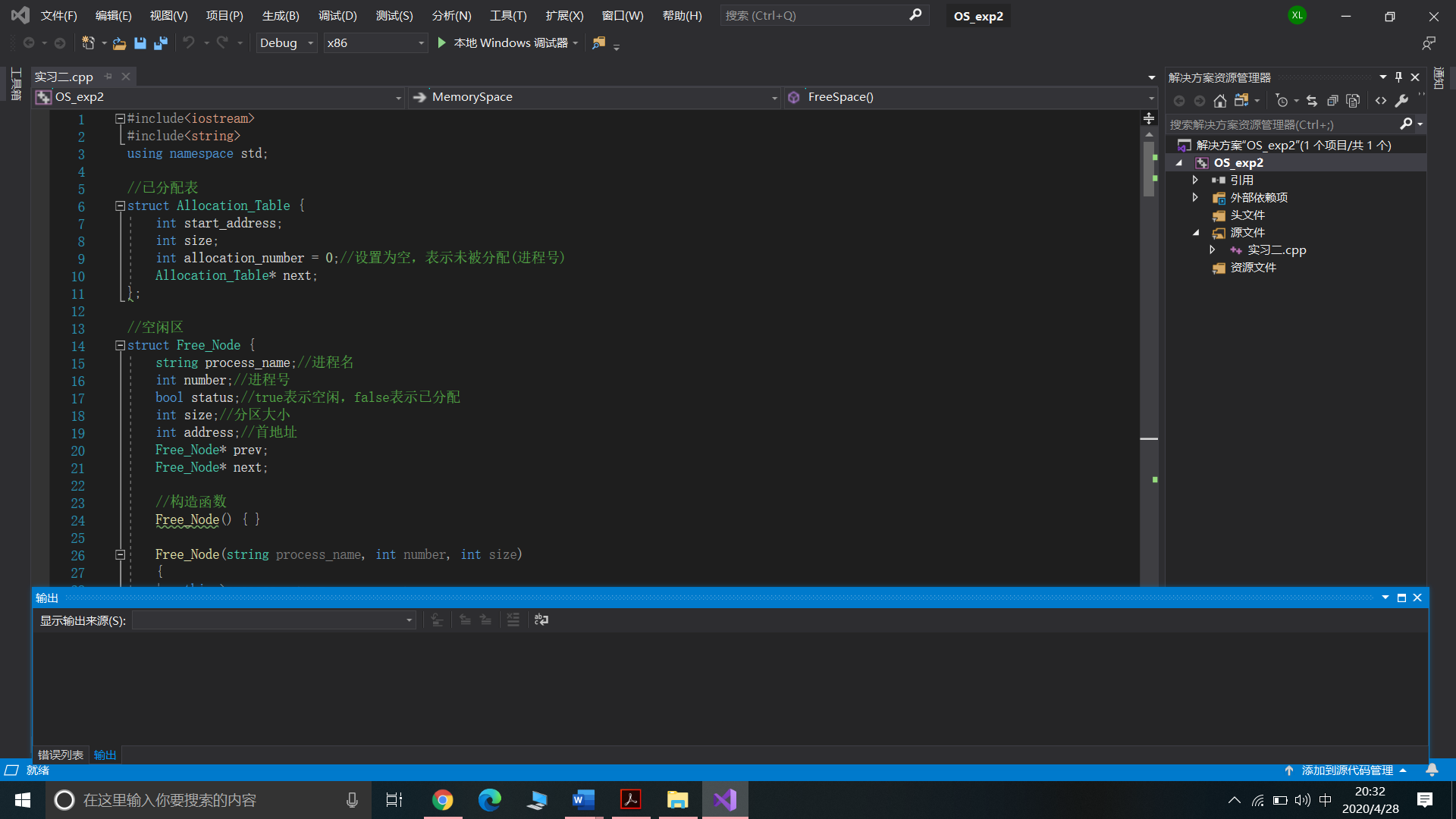
·长度——一个连续空闲区的长度；

1. 采用首次适应算法分配回收内存空间。运行时，输入一系列分配请求和回收请求。
2. 要求能接受来自键盘的空间申请及释放请求，能显示分区分配及回收后的内存布局情况。

上机实验所用平台：VS Code

**二、 数据结构及代码段分析**

数据结构为单链表（用于已分配表）和双链表（用于空闲分区）。首先建立结构体的节点信息，此部分代码如图2.1所示。



图**2.1** 数据结构

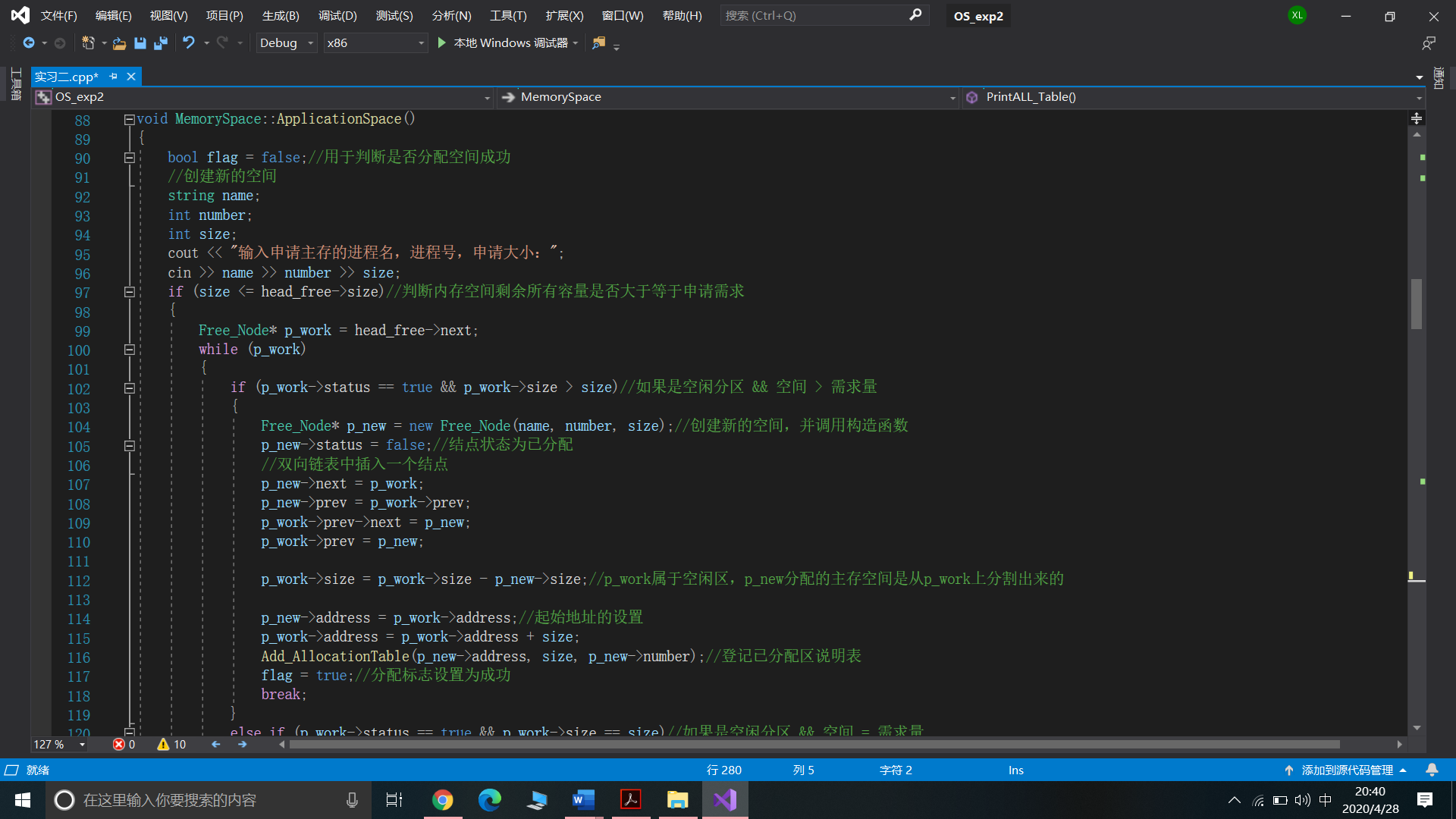
此程序主要由三部分组成，分配空间、回收空间、打印分区表信息。

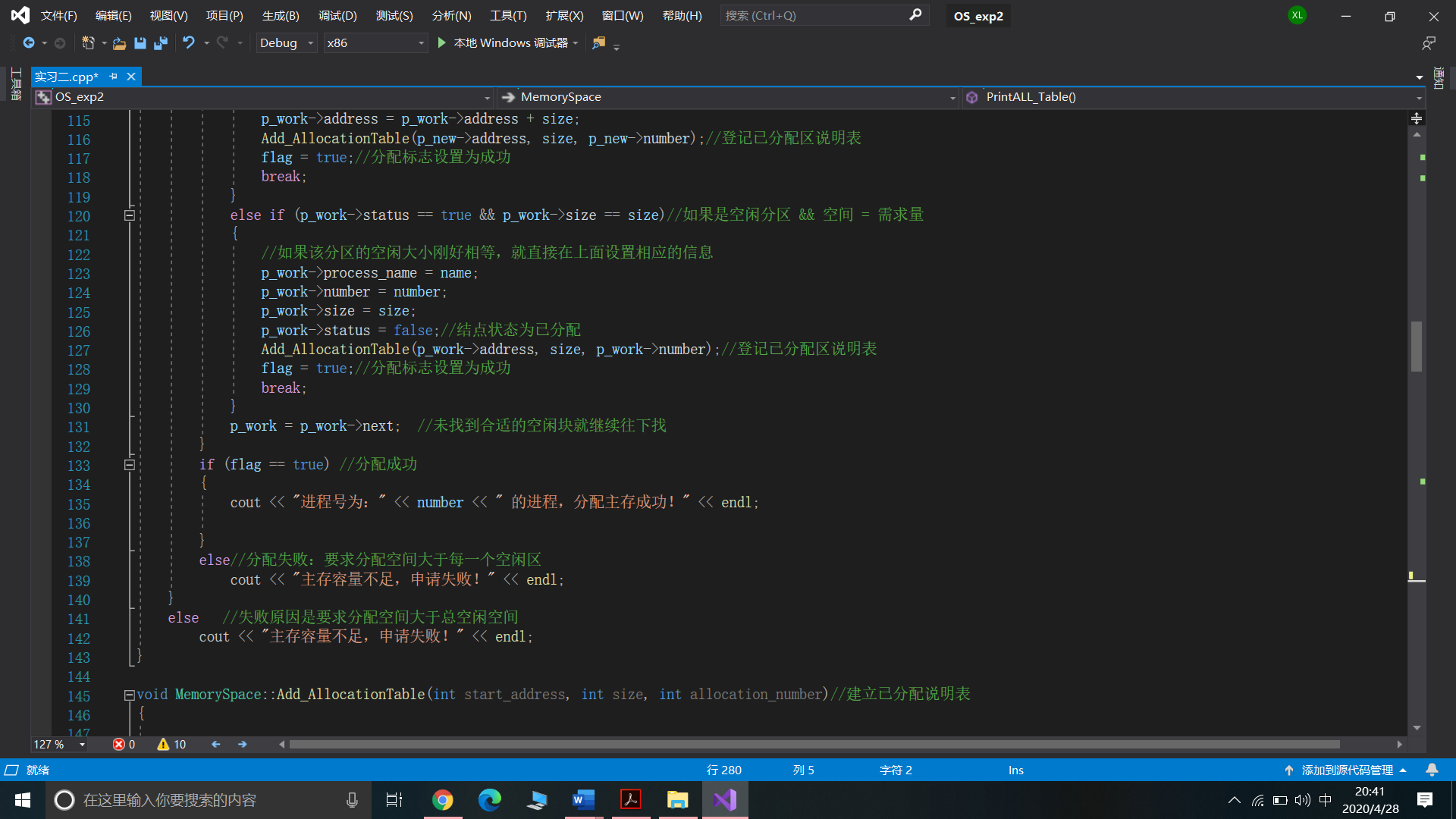
分配时设置一个bool类型的flag标志来判断空间分配是否成成功，如果申请的空间小于或等于内存剩余空间才能继续分配，否则显示“内存容量不足，申请失败”。

注意有两种分配情况：（1）当前分区状态是空闲并且当前分区的容量大于申请空间，一种是空闲分区大小大于要求的大小，此时空闲分区一分为二，一部分用来分配，另一部分重新变为新的空闲分区。将新任务的节点插入到分区表的双链表中，同时修改这两部分的状态、长度和起始地址信息。并在已分配结点表中登记该分配区域的信息（2）当前分区状态是空闲且申请容量恰好等于该分区容量，则只需要修改当前空闲分区的信息，将其状态设置为已分配，并在已分配表中登记相应信息。

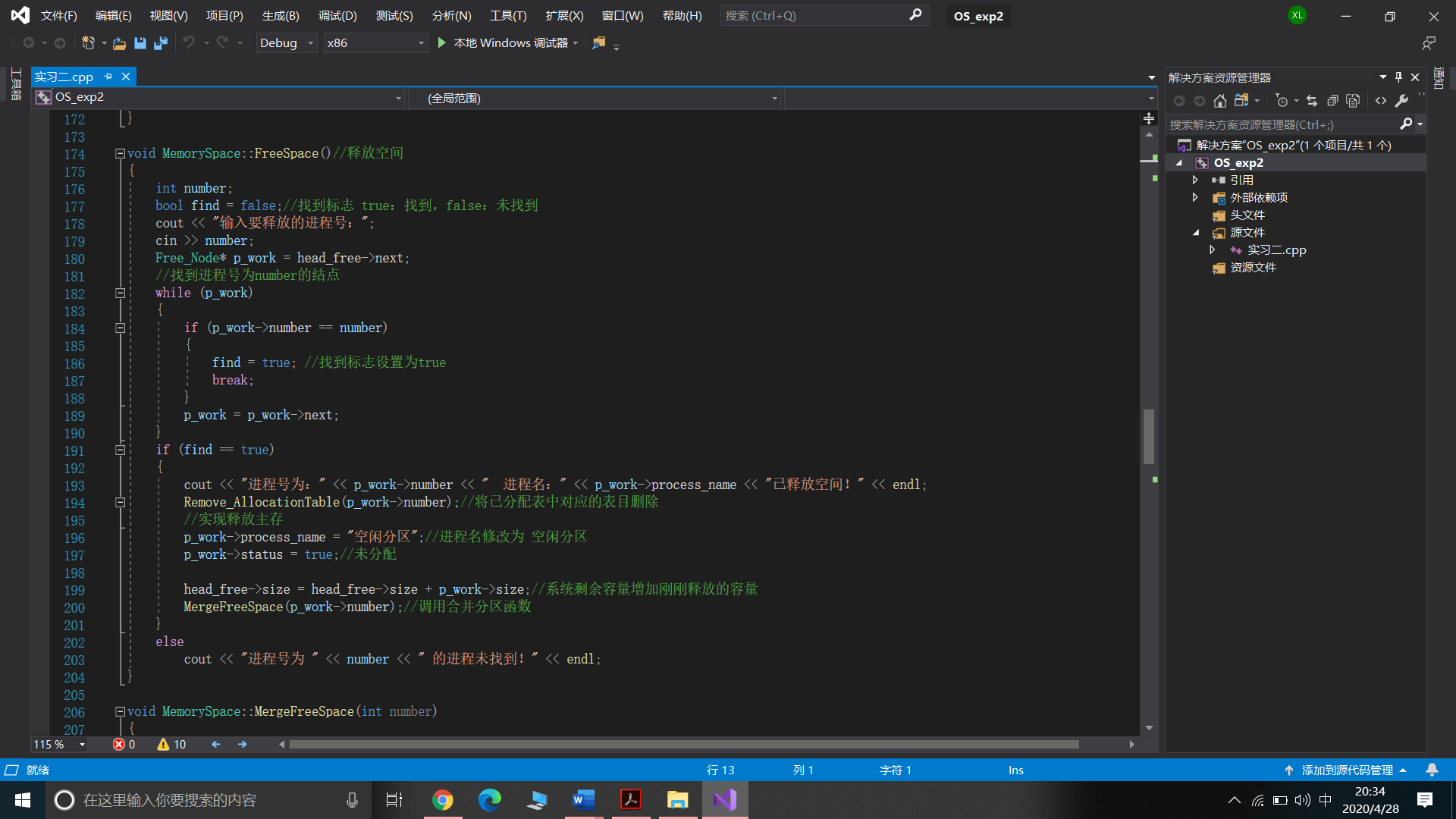
若未找到合适的空闲块就p\_work = p\_work->next顺序往下找，指导最后一个结点为止。一旦分配成功，将flag 设置为true,输出相应信息，否则分配失败，输出申请失败信息。

此部分重要代码如图2.2，图2.3所示。



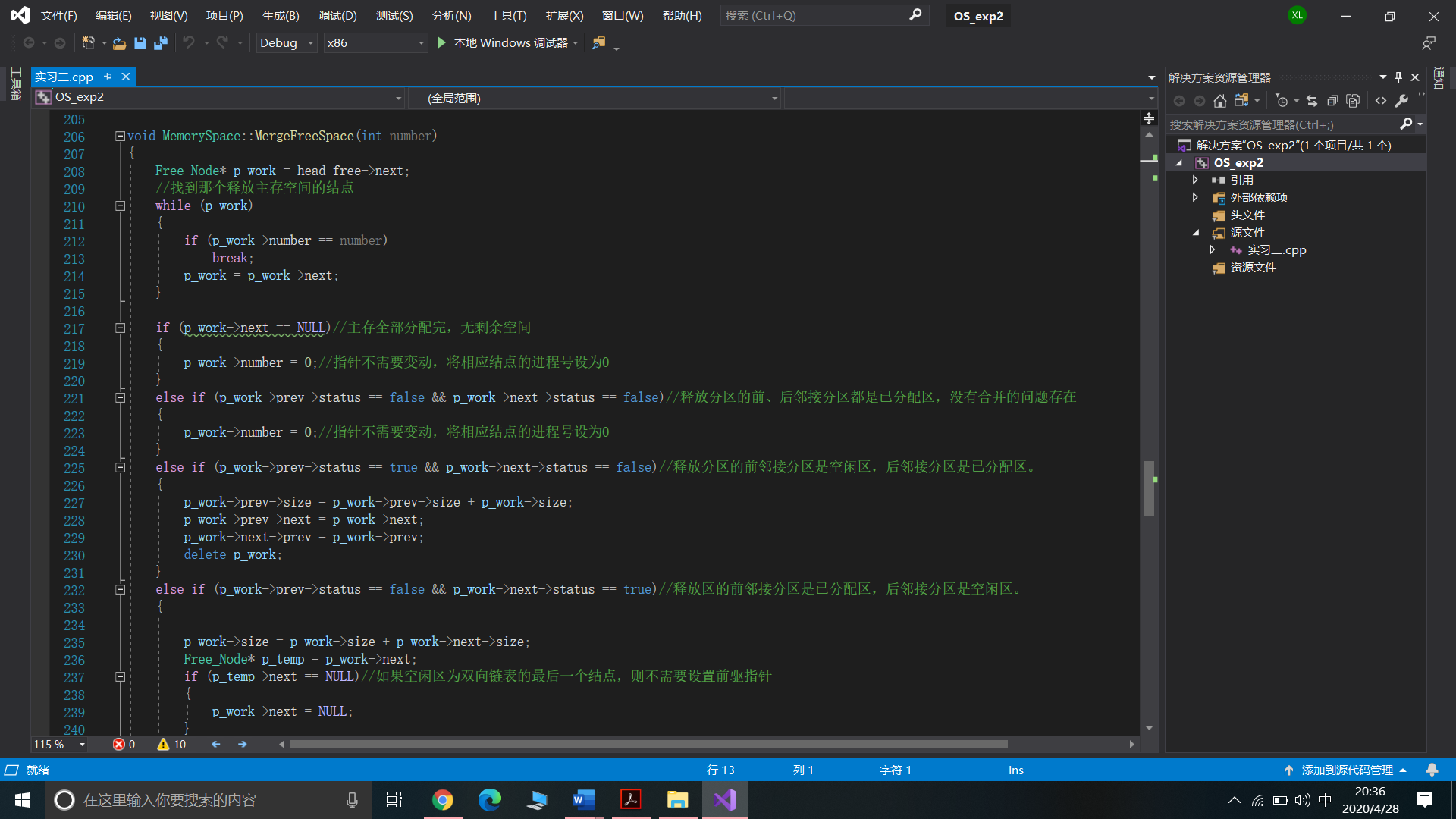
图**2.2** 分配空闲分区重要代码

图**2.3** 分配空闲分区重要代码

回收空闲分区的代码是本实验中的难点，因为要考虑到多种情况，判断所释放结点前后是否是空闲区，如何与前后的空闲区合并等问题。因此单独写了一个合并空闲分区函数，逻辑上更加清楚。回收时，依次遍历链表节点，需要判断是否有相同的任务名，若有，将空闲区size增刚刚释放的容量，并调用合并空心啊分区的函数。若没有，则find标志置false。此部分重要代码如图2.3，图2.4所示。

图**2.4** 回收空闲分区的重要代码

合并空闲分区，有四种情况：（1）释放区的前后是已分配区，无需合并，指针不需要变动，只用将相应结点进程号设置为0。（2）释放区的前邻接分区是空闲区，后邻接分区是已分配区，则将前面一块与当前释放分区合并，修改指针起始地址和分区长度，并将前一结点与后面已分配结点合并，删除当前结点。（3）释放区的前邻接分区是已分配区，后邻接分区是空闲区，则将当前释放分区和后一结点合并，修改指针起始地址和分区长度，并将前一结点与后面已分配结点合并，删除后一个结点。特别注意后邻接结点是否是最后一个结点的情况，若是则不需要设置前驱指针。（4）释放区前后都是空闲区，那么后两个空闲区与第一个空闲区合并之后删除后两个结点。



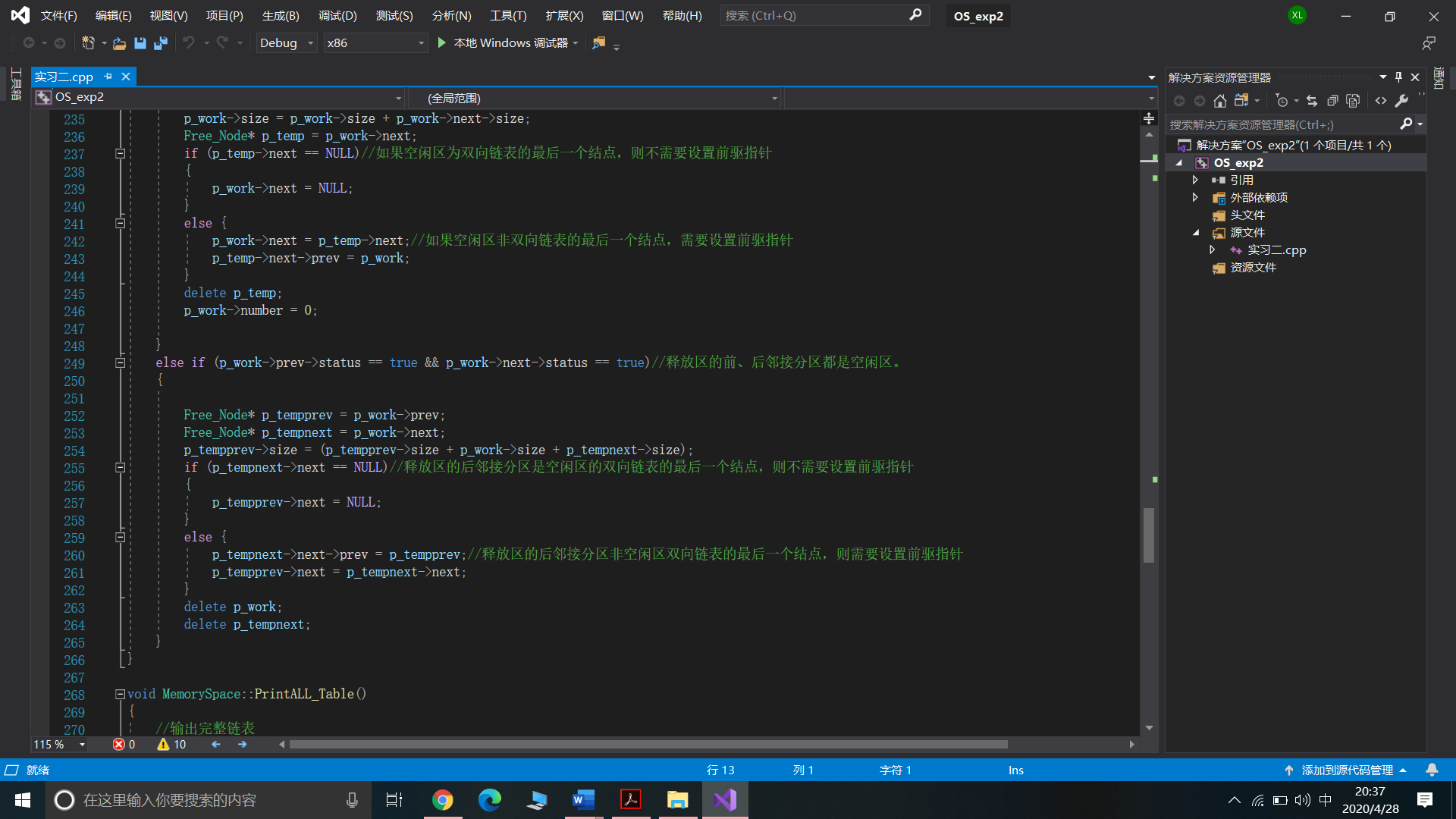
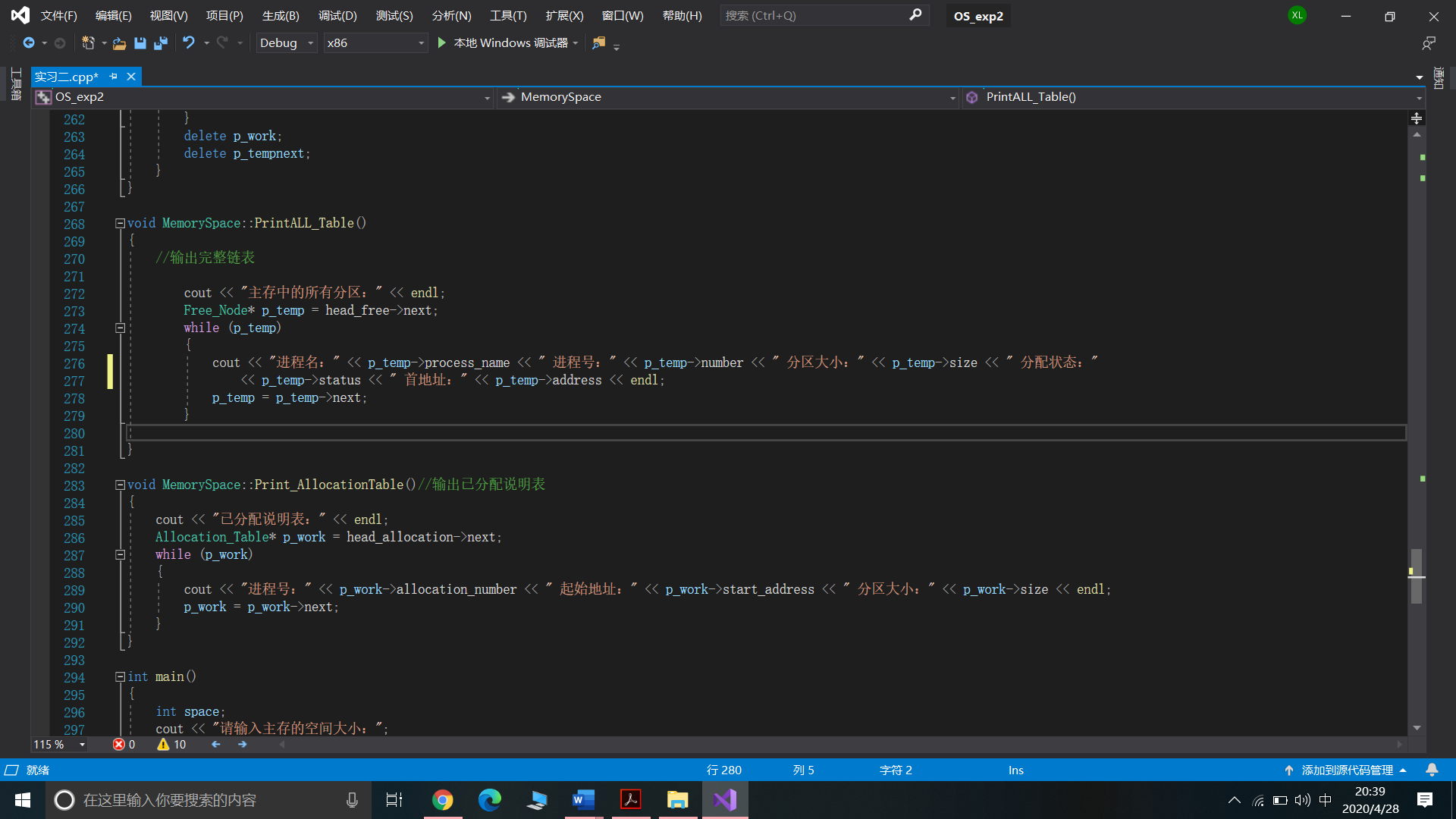
图2.5 合并空闲分区的重要代码

图2.6 合并空闲分区的重要代码

最后，即为打印分区表信息。此部分代码如图2.5所示。



图**2.5** 打印分区表信息

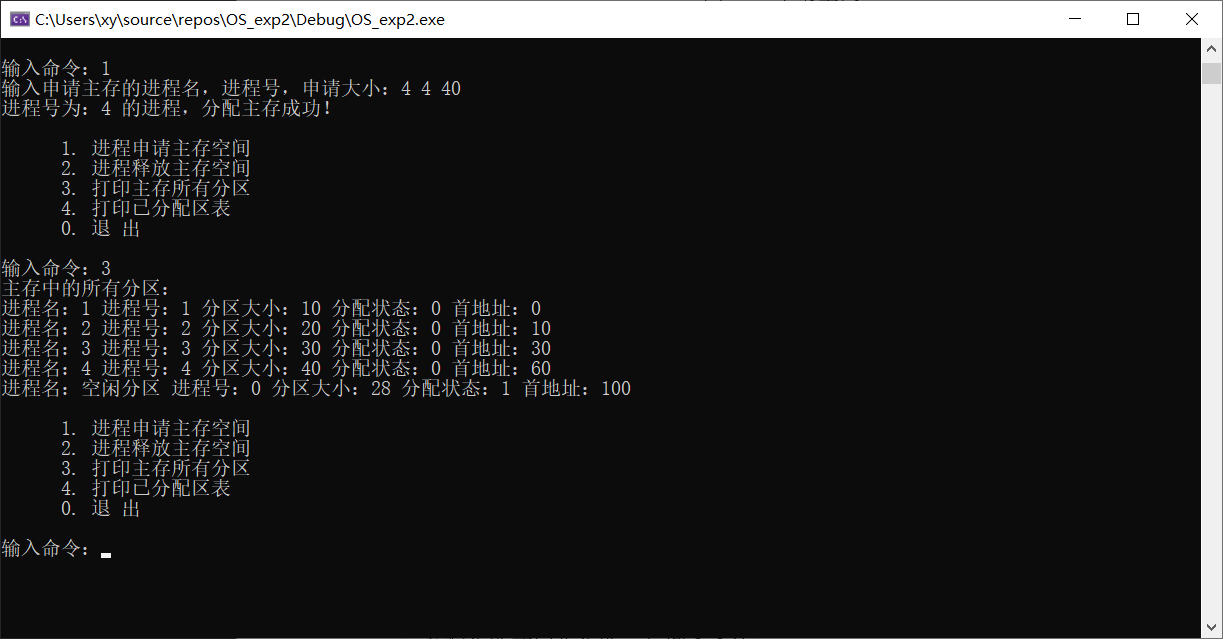
**三、 调试过程**

起始状态如图2.6所示。

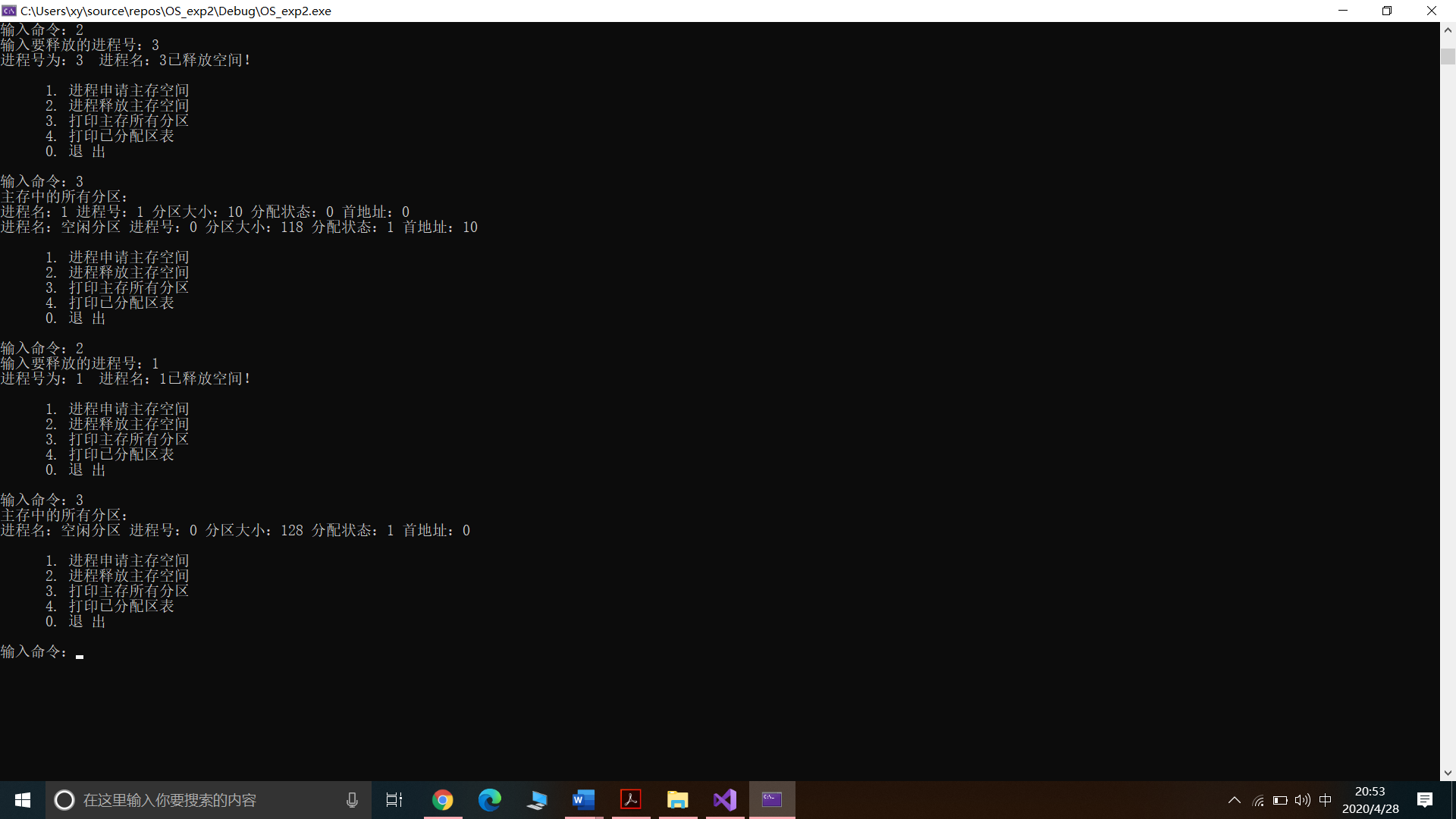


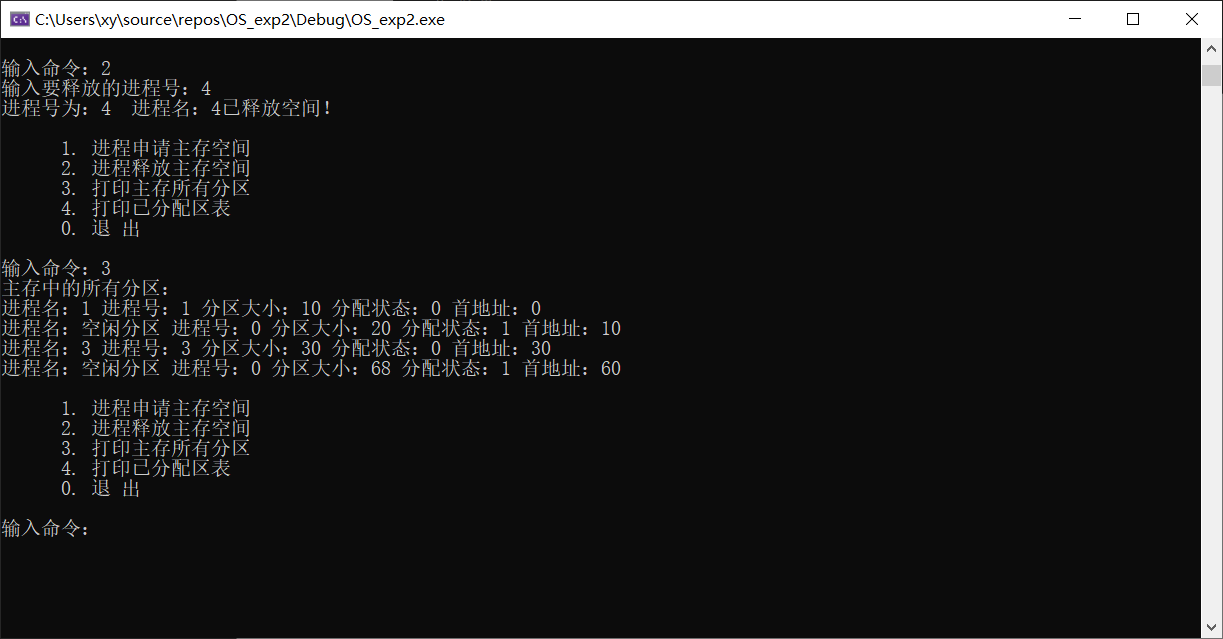
图**2.5** 运行起始状态

申请空间如图2.6所示。



图**2.6** 申请空间

回收空间如图2.7所示。



图**2.7** 回收空间

可以看到，回收后会合并上/下空闲分区。

出错处理过程中，分配不符合空间大小的数据或回收没有的任务名时，会显示分配失败或回收失败。如图2.8所示。



图**2.8** 分配空间出错处理

**四、 实验总结**

需要注意的是，分配或回收时需要修改相应节点的相关信息，例如状态、作业名等，不可遗漏。此外，回收时要注意需要合并上、下空闲分区。

通过本实习，理解了在相应的存储管理方式下应怎样进行存储空间的分配和回收。

教师评语评分

评语：

评分：

评阅人：

年 月 日

（备注：对该实验报告给予优点和不足的评价，并给出百分之评分。）