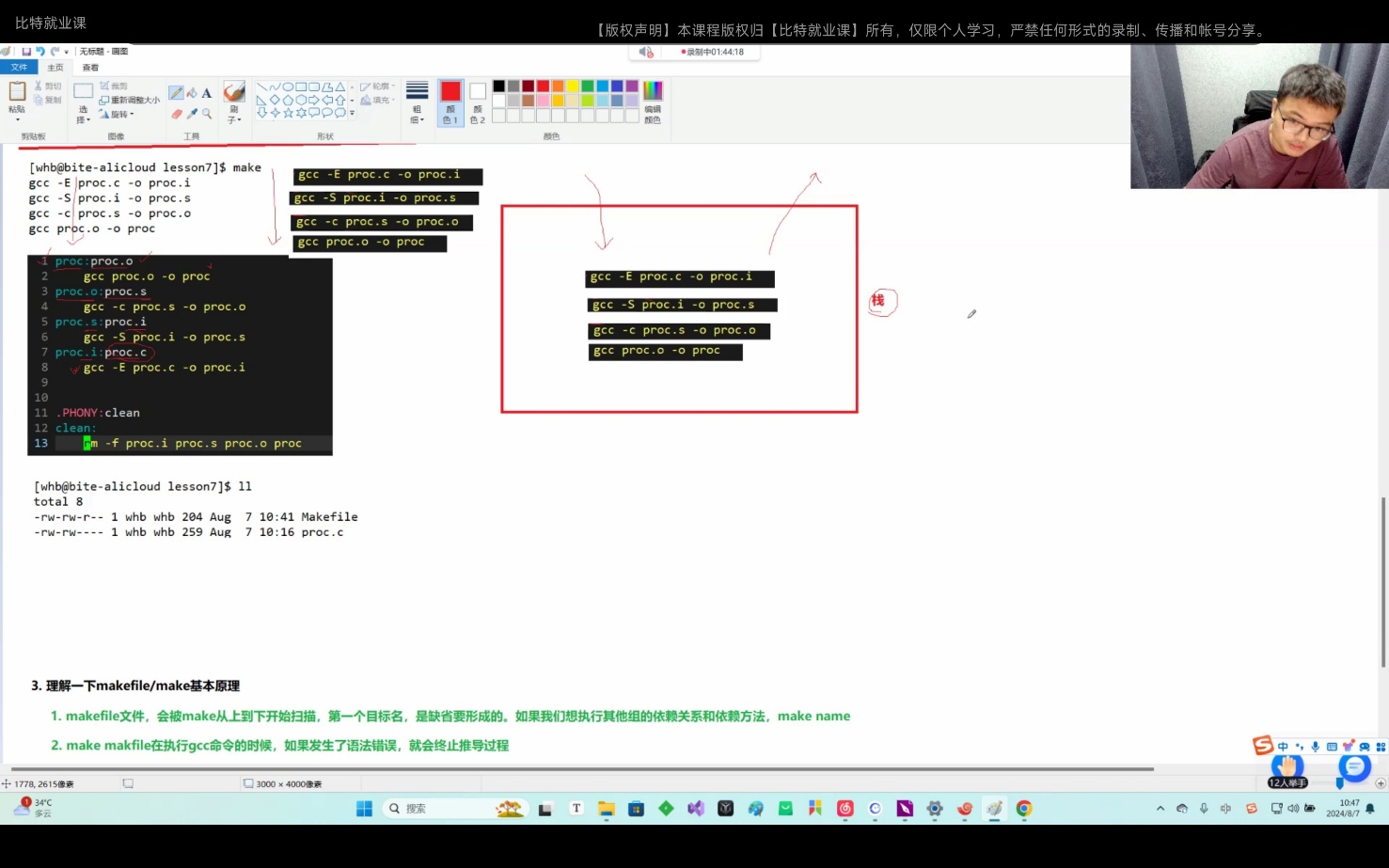
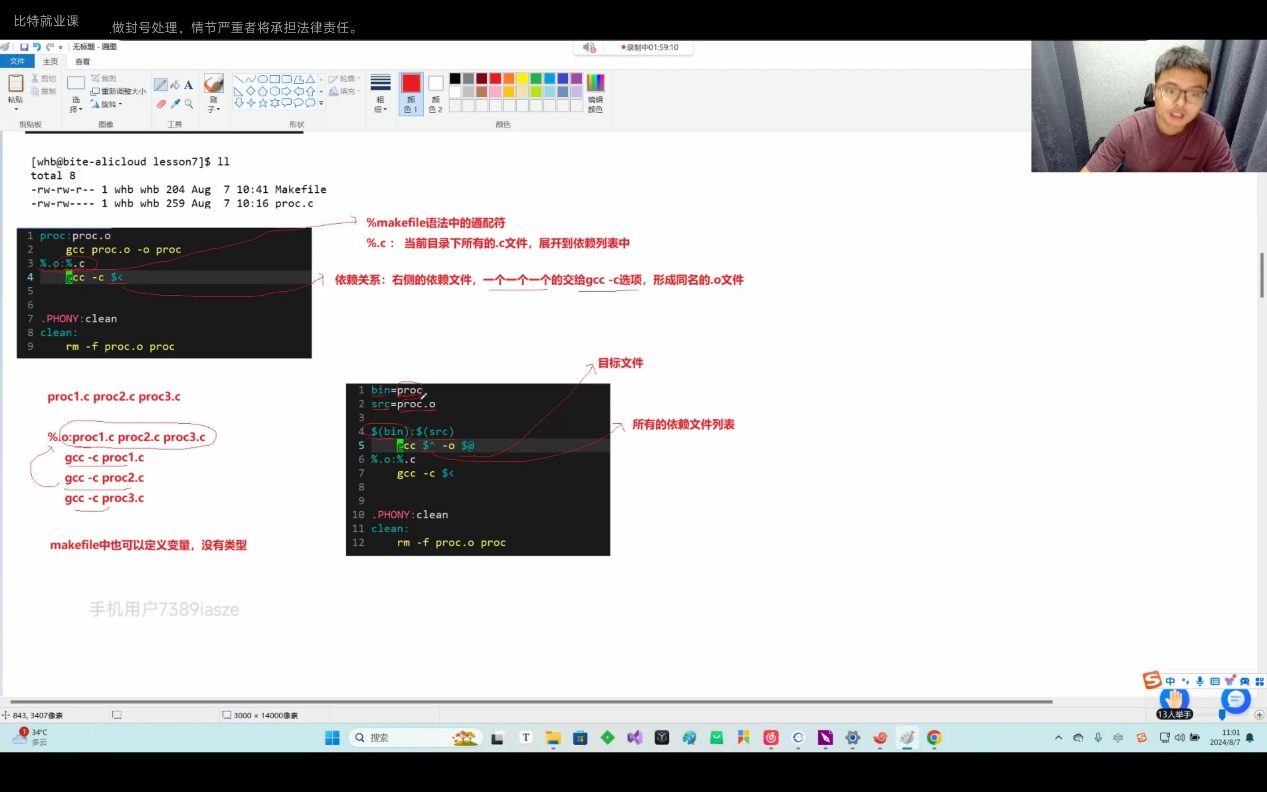
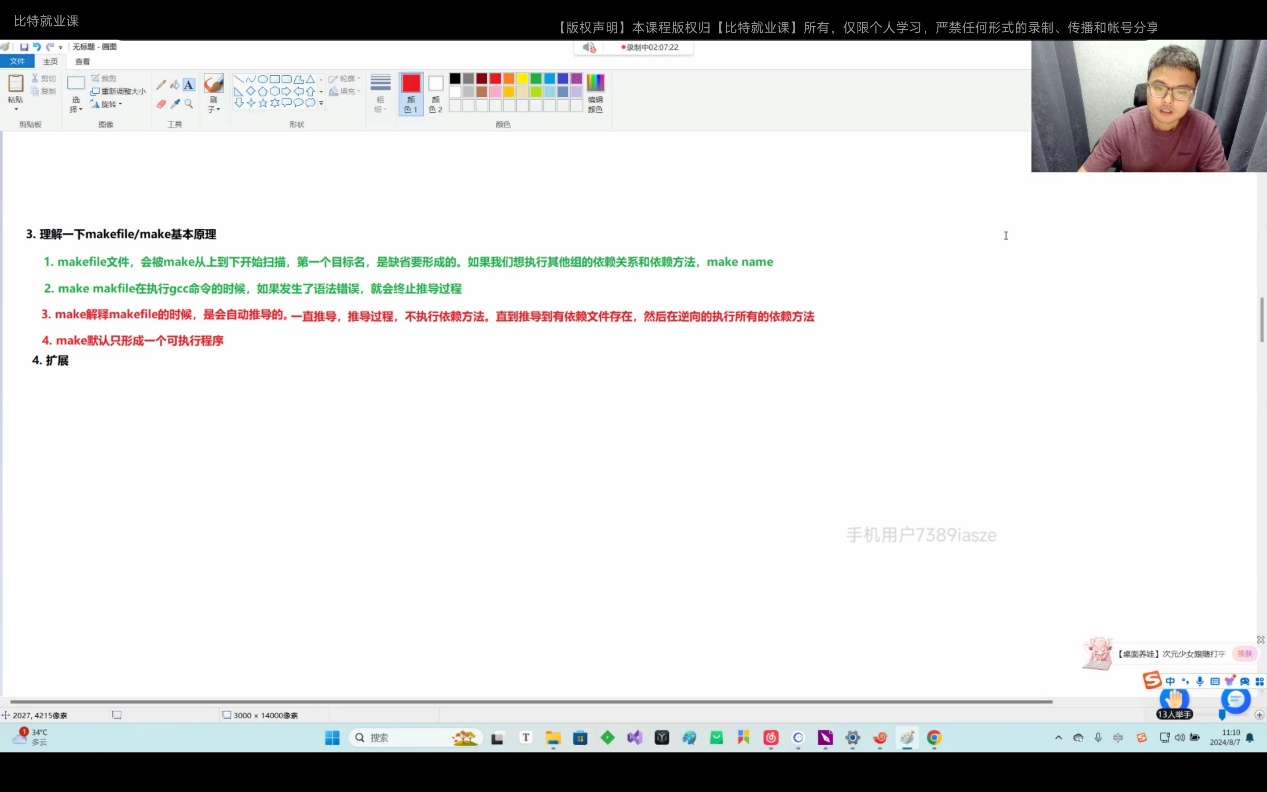
Make 在执行makefile文件的时候是会自动推导的。一直推导，推导过程，不执行依赖方法，直到推导到有依赖文件存在，然后在逆向的执行所有的依赖方法，辅助实现的是一个栈结构。

在window 编译代码形成可执行的时候，会有一个.obj的文件（目标文件）





回车和换行是一个概念吗？？

不是

回车是回到该行行首

换行是指，光标换到下一行且距离行首的距离与在上一行距离行首的距离相同。

新起一行：本质：先回车(\r)，再换行（\n） \r\n

在编程语言中

\r\n=\n

只回车 \r

换行在语言中很难。

Sleep 函数

为啥不带/n 他是sleep两秒 再printf

在这两s中他在输出缓冲区中。

\n含义为强制刷新

显示器没有类型的概念！！！显示的都是一个一个的字符。

Printf(“%d/n”,a)---》printf要把整数123，转化成’1’,’2’,’3’依次显示

因此为啥printf叫格式化输出，将各种可以输出的类型，格式化成与变量看着相同的字符串输出。显示器一直在骗我们。

C语言输出默认右对齐输出。

#program one 避免头文件重复包含

pragma once 的工作原理

​​作用机制​​：置于头文件开头，告知编译器​​该文件仅需编译一次​​。后续遇到相同文件时，编译器直接跳过包含操作。

 一、\0 的核心作用

​​标识字符串结束​​

C语言中的字符串本质是字符数组，\0 标记其逻辑终点。例如：

char str[] = "Hello";  // 实际存储：'H','e','l','l','o','\0'

函数如 printf()、strlen() 遇到 \0 即停止操作。

若缺失 \0，函数会继续读取后续内存（可能输出乱码或崩溃）。

​​占用内存但不计入长度​​

\0 占用1字节内存，但 strlen() 返回的长度不包含它。

声明数组时需预留空间：

char s[6] = "Hello";  // 正确：5字符 + 1位\0

char s[5] = "Hello";  // 错误：无空间存\0，可能越界[3,4](@ref)

⚠️ 二、常见错误场景

​​初始化时遗漏 \0​​

错误示例：

char s[] = {'a','b','c'};  // 无\0，输出乱码

printf("%s", s);           // 可能输出 "abcxxx..."（xxx为随机内存值）

正确做法：

char s[] = {'a','b','c','\0'};  // 手动添加

char s[] = "abc";               // 编译器自动添加[4,9](@ref)

​​字符串中间出现 \0​​

char s[] = "ab\0cd";

printf("%s", s);  // 输出 "ab"（\0后的内容被截断）[4](@ref)

​​数组空间不足​​

char s[5] = "Hello";  // 无空间存\0，printf() 可能越界读取[3,8](@ref)