**day01 运行最最简单的操作系统.txt**

**运行环境：windows、VMware虚拟机 / QUME**

..\z\_tools\nask.exe helloos.nas helloos.img

..\z\_tools\nask.exe：这是一个可执行文件的路径。

nask.exe 是一个汇编器（类似 NASM，但可能是一个自定义或特定版本的汇编器），用于将汇编代码编译成机器码。

路径中的 ..\z\_tools 表示从当前目录向上一级，然后进入 z\_tools 文件夹，找到 nask.exe。

生成的img文件，无法用 Windows 自带的资源管理器、虚拟光驱软件（如 Daemon Tools）或常规的镜像挂载工具来“打开”它并看到里面的文件。这些工具期望看到的是标准的光盘文件系统（如 ISO9660），而创建的映像不包含这些结构。

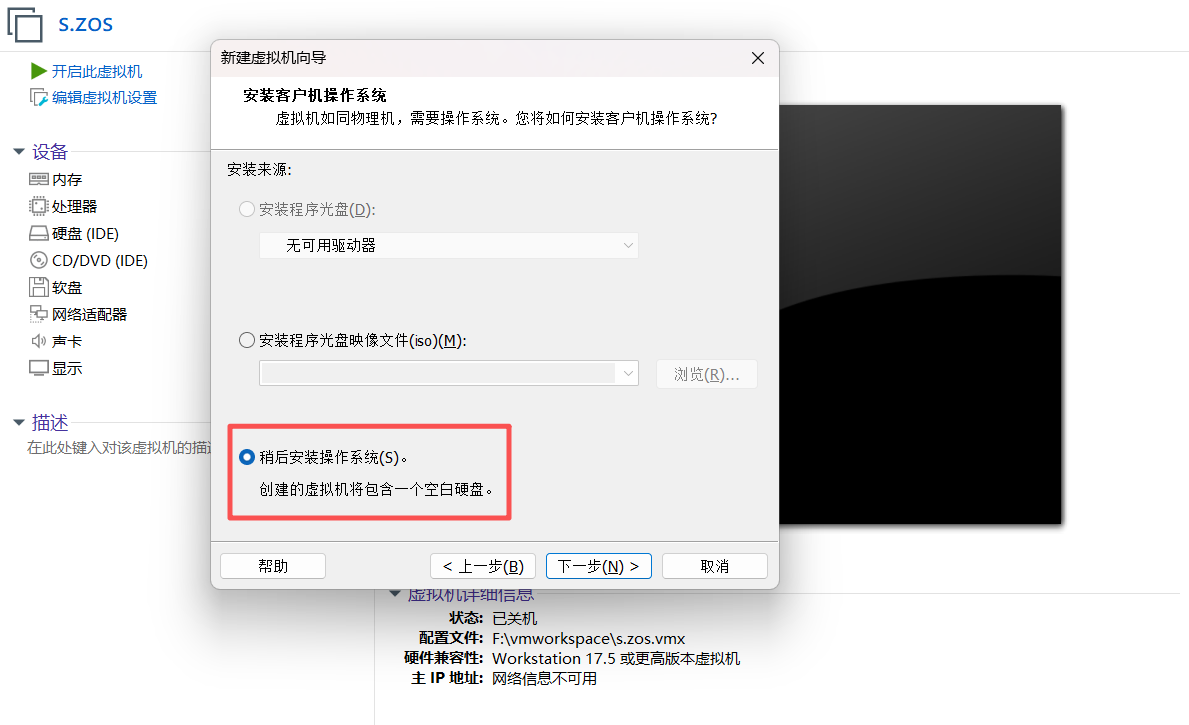
**为什么说它不是标准的光盘映像？**

**原始二进制映像（Raw Binary Image）**：由 nask.exe 生成的 helloos.img 是一个**原始的、未格式化的二进制块**。它通常被设计成**软盘（Floppy Disk）的精确镜像**。在早期计算机或操作系统开发中，软盘是常见的启动媒介。

**引导扇区**：您提供的十六进制数据开头是 EB4E90，这是一个典型的 x86 CPU **跳转指令**。这意味着这个映像的第一个扇区（512字节）是一个**引导扇区（Boot Sector）**。计算机的 BIOS 或虚拟机在启动时，会直接将这个扇区加载到内存并执行其中的机器指令，它不依赖于任何操作系统或文件系统来“解析”它。

**无标准文件系统**：虽然您的引导扇区里有一个字符串 FAT12，但这很可能只是一个占位符或为了兼容性而写的，整个映像（通常只有 1440KB，即 1.44MB 软盘的大小）可能并没有真正格式化出一个完整的、可被 Windows 识别的 FAT12 文件系统。它里面包含的只有您用汇编代码写的程序二进制码，可能后面跟着一些数据，但没有目录表、文件元数据等东西。

**第一种方式，我们利用VMware，建立一个新的操作系统，**





添加一个新的软盘，并导入我们刚刚写入helloos.img（或者asm.bat 生成的）文件。从而可以启动我们所新的建立操作系统！！



**第二种方式，将编译好的镜像文件部署到模拟器（QEMU）并启动模拟器的自动化脚本命令**

copy helloos.img ..\z\_tools\qemu\fdimage0.bin

..\z\_tools\make.exe-C ../z\_tools/qemu

**命令一：copy helloos.img ..\z\_tools\qemu\fdimage0.bin**

这条命令的作用是**复制并重命名**镜像文件。

**copy**: Windows 系统的文件复制命令。

**helloos.img**: 这是源文件，即您刚刚通过 nask.exe 编译生成的原始磁盘映像文件。

**..\z\_tools\qemu\fdimage0.bin**: 这是目标路径和文件名。

..\z\_tools\qemu\: 表示从当前目录向上一级，然后进入 z\_tools 文件夹下的 qemu 子目录。

fdimage0.bin: 这是目标文件名。

**为什么这么做？**

模拟器（在这个例子中是 QEMU）在启动时，通常会从一个**预定义的文件名**（例如 fdimage0.bin）读取软盘映像。这条命令的作用就是将您新编译好的 helloos.img **“安装”或“部署”** 到模拟器期望的固定位置和文件名上，这样模拟器就知道该加载哪个文件了。

可以把它想象成：“把刚烧录好的系统光盘（helloos.img）放到一个指定的、唯一的光驱里（fdimage0.bin）”。

**命令二：..\z\_tools\make.exe -C ../z\_tools/qemu**

这条命令更复杂一些，它使用 make 工具来执行一个构建任务。

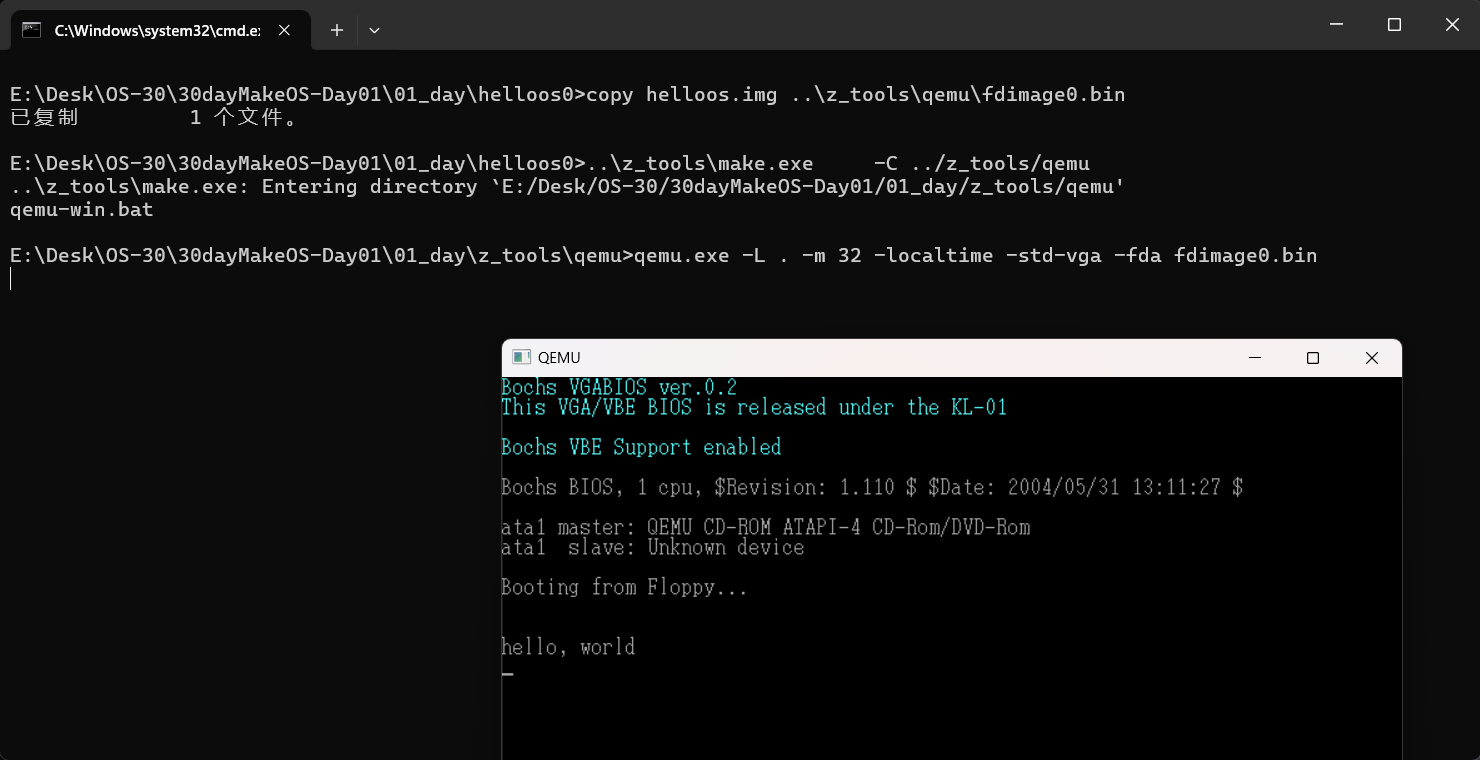
**..\z\_tools\make.exe**: 调用 z\_tools 目录下的 make.exe 程序。make 是一个经典的构建自动化工具，它通过读取名为 Makefile 的脚本文件来执行一系列预定义的命令。

**-C ../z\_tools/qemu**: 这是 make 命令的参数。

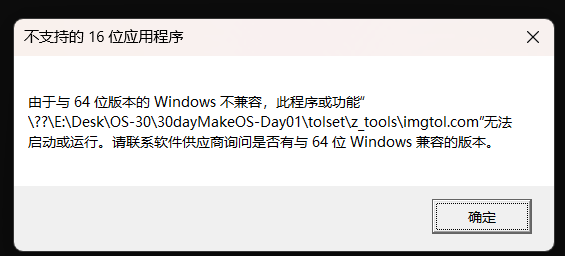
-C 参数的意思是 “Change directory”，即告诉 make.exe：**“请先切换到 ../z\_tools/qemu 这个目录，然后再开始你的工作”**。

**这条命令实际会做什么？**

它切换到 ../z\_tools/qemu 目录后，会寻找该目录下的 Makefile 文件。这个 Makefile 里面预先写好了启动 QEMU 模拟器的命令和参数。



这条命令 ..\z\_tools\imgtol.com w a: helloos.img 是一个**用于将磁盘映像文件写入物理软盘**的指令。这在过去（2000年代初期及更早）是开发和测试引导程序、操作系统内核的常见方法。（这个与我们现在的windows不兼容了，所以推荐使用第一种）



**完整的开发循环 (Development Loop)**

**编辑源代码 (zsos.nas)**

您使用文本编辑器（如 Visual Studio Code, Notepad++ 等）修改 zsos.nas 文件。

这个文件里写的是 **x86 汇编语言**。您可以通过修改这些指令来改变您“操作系统”的行为，例如显示不同的文字、绘制不同的图形、处理键盘输入等。

**编译 (asm2.bat)**

您运行批处理文件 asm2.bat，它内部执行了 ..\z\_tools\nask.exe zsos.nas zsos.img。

**nask.exe** 是专门为这类教学项目设计的汇编器。它的作用就是将人类可读的汇编指令 (zsos.nas) **翻译（汇编）** 成计算机可以直接执行的**机器码**，并将这些机器码按照特定的格式（如引导扇区、FAT12文件系统）打包成一个完整的磁盘映像文件 zsos.img。

您得到的 zsos.img 文件本质上就是一个包含二进制机器码的软盘映像。

**运行 (run2.bat)**

您运行另一个批处理文件 run2.bat。根据之前的讨论，这个 run2.bat 的内容很可能就是：

batch

copy zsos.img ..\z\_tools\qemu\fdimage0.bin

..\z\_tools\make.exe -C ../z\_tools/qemu

这条命令将新编译好的映像文件复制到 QEMU 模拟器期望的位置，然后自动启动 QEMU。

**QEMU** 会模拟出一台“虚拟电脑”，把这整个 zsos.img 文件当作一张**虚拟软盘**，并尝试从它启动。这时，您写在 zsos.nas 里的代码就开始在这台虚拟电脑上运行了！