第四次作业

程远 2234412848

1 第一题

第一步，设计程序：程序员用一种或多种语言编写源码。例如只用 c 和 c++ 编写的简单程序，或者是再加入 c# 等语言共同编写一个桌面应用。

第二步，编译源码：在这一步中，人类能够被阅读的源代码将会最终被编译为机器码，下面以 try.cpp 的编译过程为例解释。首先、try.cpp 会经过预处理步骤，这一步骤包含处理所有带有 # 的语句，包括 #include、#define 和 #ifdef 等等。这一过程中编译器会把头文件的内容插入到指令所在位置，替换定义的宏，完成条件编译等，最终生成 try.i 文件。在这之后，通过编译，将预处理后的代码转换为汇编代码，生成 try.s 文件。之后进行汇编，这一过程会解析汇编代码中的指令和操作数，将其转换为机器语言指令，处理代码符号，最终生成目标代码 try.o。最后是链接，链接器在此时将多个目标文件和库文件连接到一起，生成最终的可执行文件 try.exe。

第三步，储存：将生成的可执行文件储存在计算机的磁盘上。

第四步，程序加载：这一过程比较复杂，大致分为以下几步。a. 文件系统访问、操作系统通过特定的文件系统在磁盘中找到 try.exe 并读取其内容。b. 分配内存、操作系统为程序分配内存空间，并将程序的代码加载到内存中，通过调度算法让进程分时分段执行。c. 创建进程、程序一旦被装入内存执行，就被称作进程。进程中有程序代码、程序计数器、堆栈段和数据段等信息。每个进程都有独立的内存空间和资源。d.CPU 调度、通过调度算法（如时间片轮转、优先级调度等）确保 CPU 时间被多个进程公平且高效地使用 。当然，也可以手动指定优先级，让某些进程优先运行。e. 程序执行、CPU 按照取指、解码、执行的顺序依次执行程序中的所有指令。

第五步，中断处理：当设备收到输入型号时，便通过控制线向 CPU 发送一个中断信号，CPU 收到终端信号后，再通过中断响应程序进行处理。这能使得 IO 过程与 CPU 并行工作，仅当输入完成时才需要进行终端处理。

第六步，程序结束：当程序执行完毕或被用户终端后，操作系统会回收程序占用的内存与句柄等，并将其从进程列表中移除。

2 第二题

操作系统本质上是一组“管理各种资源以便执行应用程序”的程序，即管理计算机硬件与软件资源，以为应用程序提供服务。这种管理可以分为“分工”和“合作”两部分。“分工”是指独立管理各个部件，例如磁盘、内存、进程、图形界面等。而实现分别管理后，就需要各部分的“合作”。此时操作系统会管理各部分合作完成“让 CPU 执行存储在外存上的程序”这一任务。例如通过管理程序装载到内存上的过程来实现内存管理，通过规定程序如何被 CPU 执行来进行进程管理。

操作系统将程序载入内存主要步骤如下。拿启动一个 exe 程序来举例，当用户双击一个 exe 文件后，操作系统收到启动程序的请求，开始加载程序。此时操作系统先通过内存管理模块，确定程序需要的内存大小，并为其分配一定的内存空间。内存分配好后，操作系统便将 exe 文件通过文件管理模块读取到内存当中，并初始化程序环境（设置堆栈、初始化数据段等）。在这之后，OS 会为正在运行的程序创建一个进程，并通过时间片轮转等方法调度内存。

操作系统将自身载入内存的过程如下。第一步为加点自检（Power-On Self Test），简称 POST。当计算机通电启动时，BIOS 首先启动 POST，检查内存、CPU、GPU 等基本硬件是否正常工作。第二

步为查找启动设备。自检完成后，BIOS 会查找硬盘、光盘、USB 设备并从这些设备的引导扇区加载启动程序。随后进行第三步，加载引导程序。找到启动设备后，BIOS 将控制权移交给引导程序（Boot Loader），引导程序储存在设备的引导扇区，用于将操作系统的内核加载到内存中。第四步为执行引导程序。第一阶段的引导程序储存在主引导记录中，第一阶段的引导陈程序将被加载并执行第二阶段的引导程序（第二阶段的引导程序储存在磁盘的特定分区中）。随后是第五步，内核加载。第二阶段引导程序将内核文件从磁盘加载到内存中，内核被成功加载后，控制权便会从引导程序被移交给操作系统内核。随后内核开始初始化，设置内存管理单元与初始化诸如进程管理、文件系统这样的子系统。内核初始化后将启动系统进程，如 Windows 系统中的 smss.exe。最后操作系统会启动用户界面，提供登录服务。通过以上的过程，操作系统成功将自身加载到了内存当中。