操作系统概述

- 操作系统做了什么？
- 操作系统的定义与作用
- 操作系统的主要特征
- 典型操作系统的架构
- 操作系统的分类

它们之间有哪些关联，最后我们介绍一下操作系统的分类

操作系统做了什么？(1/4)

```
#include <stdio.h>
int main(int argc, char *argv[])
   {
        puts("hello world");
        return 0;
   }
```

world" 好下面我们来看一下这个程序的执行过程

操作系统做了什么？(2/4)

- 用户告诉操作系统执行helloworld程序(如何告知？)
- 操作系统：找到helloworld程序的相关信息，检查其类型是否是可执行文件；并通过程序首部信息，确定代码和数据在可执行文件中的位置并计算出对应的磁盘块地址（文件格式？）

那么我们知道Windows操作系统下，可执行文件的格式呢是PE格式

操作系统做了什么？(2/4)

- 用户告诉操作系统执行helloworld程序(如何告知？)
- 操作系统：找到helloworld程序的相关信息，检查其类型是否是可执行文件；并通过程序首部信息确定代码和数据在可执行文件中的位置并计算出对应的磁盘块地址（文件格式？）

在Linux操作系统下，可执行文件的格式呢是ELF格式

操作系统做了什么？(2/4)

- 用户告诉操作系统执行helloworld程序(如何告知？)
- 操作系统：找到helloworld程序的相关信息，检查其类型是否是可执行文件；并通过程序首部信息，确定代码和数据在可执行文件中的位置并计算出对应的磁盘块地址（文件格式？）
- 操作系统：创建一个新的进程，并将helloworld可执行文件映射到该进程结构，表示由该进程执行helloworld程序
- 操作系统：为helloworld程序设置CPU上下文环境

准备执行这条这个程序，那么下一个指令周期就是执行helloworld程序了

操作系统做了什么？(3/4)

- 执行helloworld程序的第一条指令，发生缺页异常
- 操作系统：分配一页物理内存，并将代码从磁盘读入内存，然后继续执行helloworld程序
- helloworld程序执行puts函数（系统调用），在显示器上写一字符串
- 操作系统：找到要将字符串送往的显示设备，通常设备是由一个进程控制的，所以，操作系统将要写的字符串送给该进程

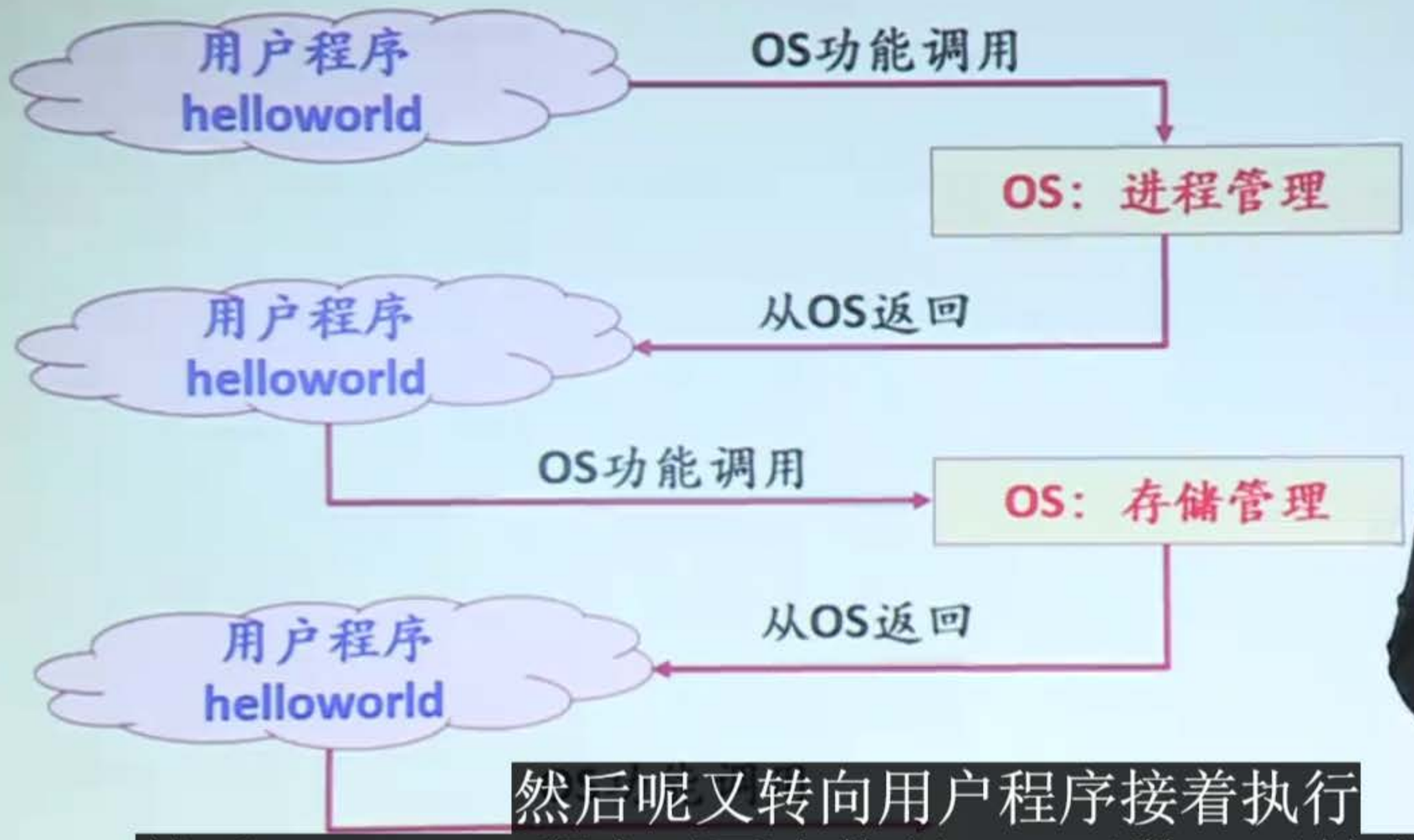
把要写的字符串实际上是送给了这个进程

操作系统做了什么？(4/4)

- 操作系统：控制设备的进程告诉设备的窗口系统它要显示字符串，窗口系统确定这是一个合法的操作，然后将字符串转换成像素，将像素写入设备的存储映像区
- 视频硬件将像素转换成显示器可接收的一组控制数据信号
- 显示器解释信号，激发液晶屏
- OK！！！我们在屏幕上看到了“hello world”

从启动到执行，到结束 从这个过程当中，我们得到什么结论呢？

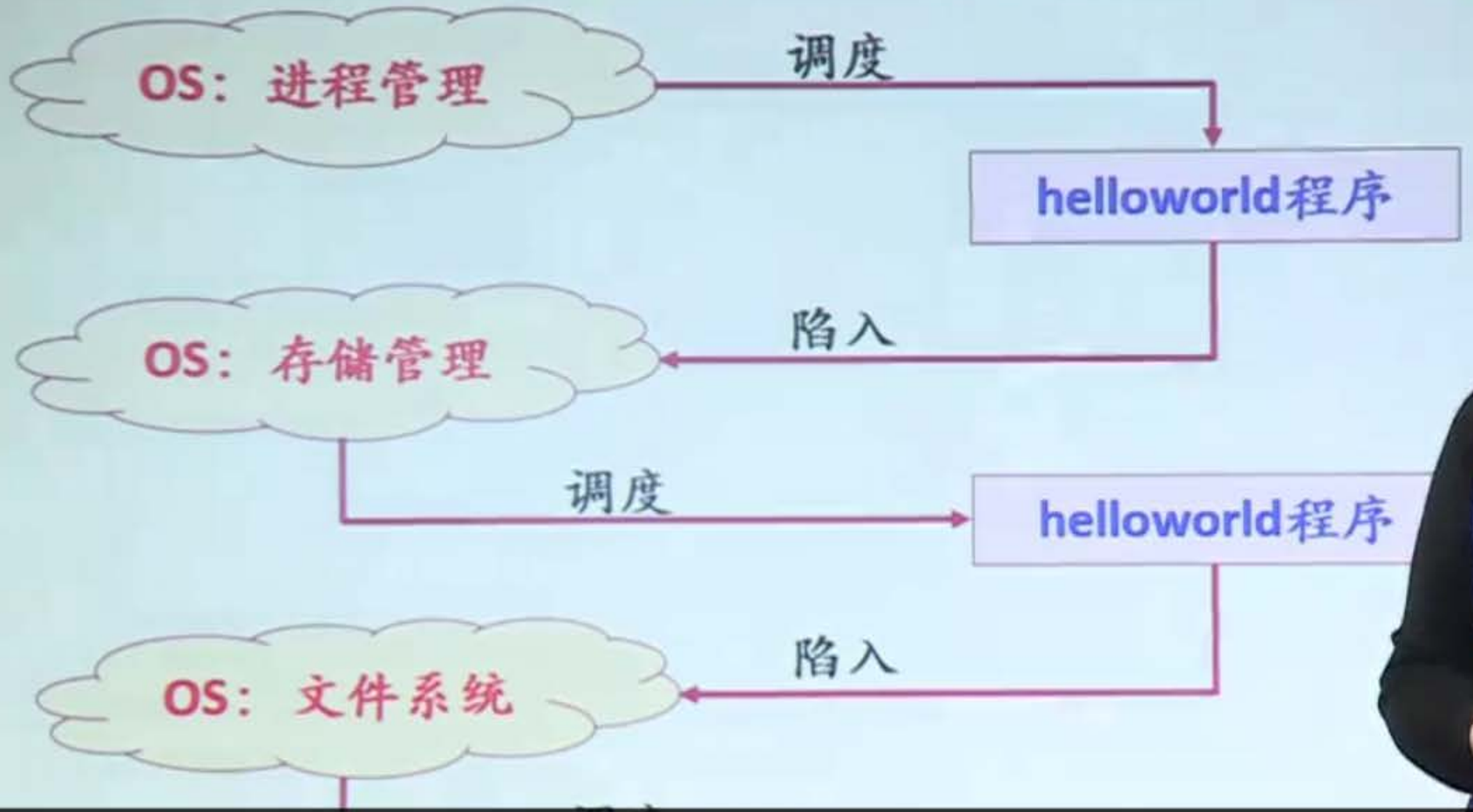
从上述步骤中得到什么？



然后呢又转向用户程序接着执行

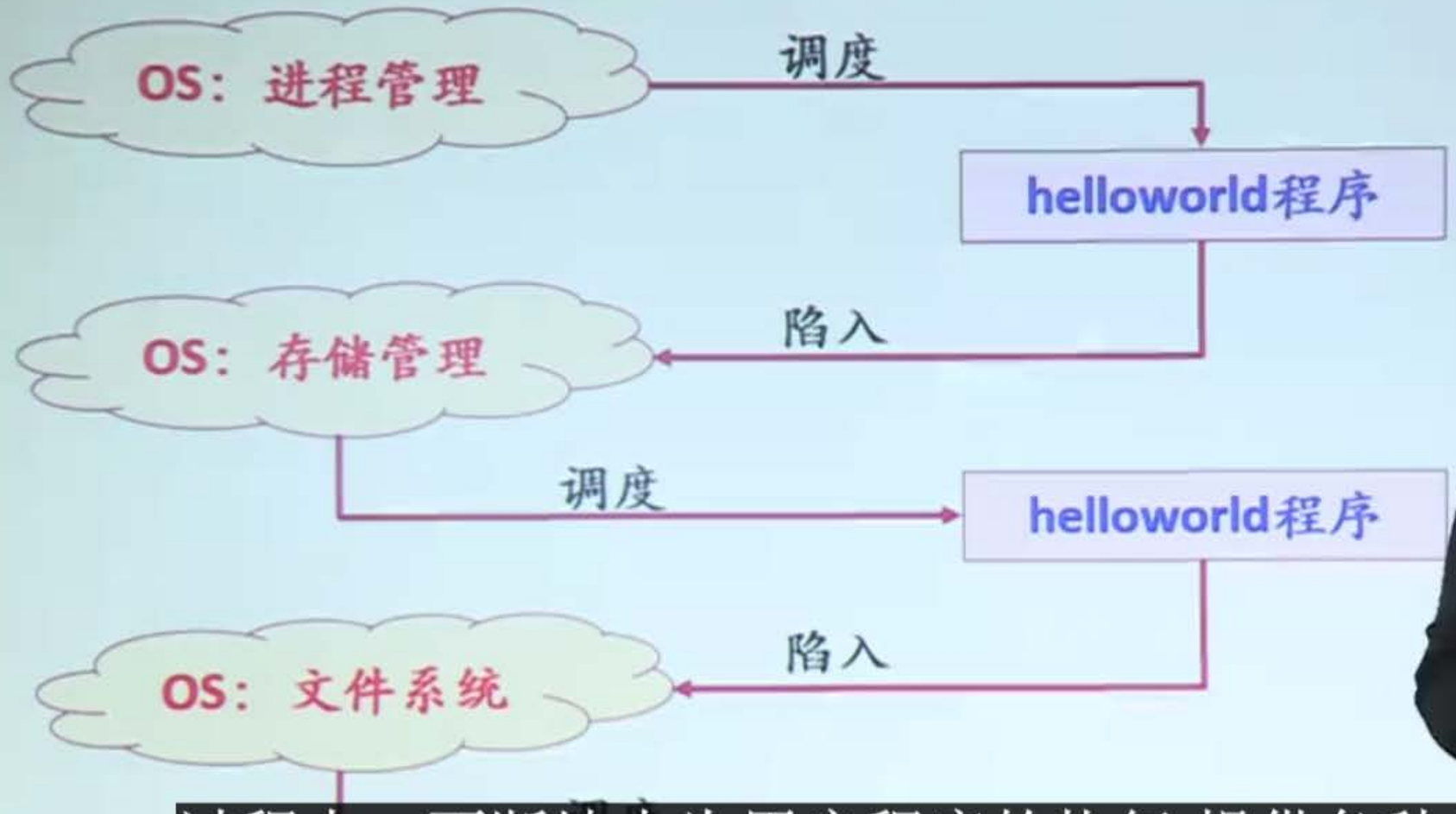
然后helloworld程序又要去执行puts函数，又转向了操作系统

换个角度看用户程序的执行



执行过程中，操作系统负责了这个程序的启动过程 负责了这个程序执行的过程，
同时在执行的

换个角度看用户程序的执行



过程中，不断地去为用户程序的执行 提供各种各样的支持

0:00 大家好，今天我给大家带来的是操作系统原理课的第一讲 操作系统概述，在这一讲里头，我们将学习以下这些内容 首先我们通过一个程序的执行过程 来看一下，操作系统做了什么，然后我们介绍操作系统的定义和它的作用 操作系统是一个软件，那么相对于其它的软件 它有哪些特征呢？有哪些特点呢 另外我们介绍一些典型的操作系统架构 通过介绍Windows、 Unix和Linux操作系统的架构 让我们有一个感性的认识，看一看操作系统都有哪些功能 它们之间有哪些关联，最后我们介绍一下操作系统的分类 首先我们先看一下 操作系统做了什么，这是一个 简单的C语言程序helloworld 我们想通过这个程序的执行过程，来看一下操作系统 对这个程序的执行过程有哪些支持，我们先来看一下程序 这个程序里头主要功能是调用了一个puts函数 它的作用是在标准的输出设备上，也就是我们说的显示器上 显示字符串"hello world" 好下面我们来看一下这个程序的执行过程 首先我们要启动程序执行 用户要告诉操作系统，执行helloworld程序 大家可以想一想，有哪些方式 可以告诉操作系统要执行某个程序 比如说，我们可以在命令行键入相应的命令 我们也可以通过鼠标双击helloworld程序的图标 通常告诉操作系统执行一个程序的方法 不止一个，操作系统接到了用户的请求之后 就会根据用户提供的文件名 到磁盘上找到这个程序的相关信息 找到信息之后，会去检查这个程序 是不是一个可执行文件，因为有的时候我们 的一个文件它不是可执行文件，那么操作系统就不能执行这个文件 就会去报错，那么如果是一个可执行文件，操作系统才能正确地执行 在检查完类型之后，操作系统会根据程序首部信息 来确定代码和数据在这个可执行文件当中的位置 并计算出它相应的磁盘块地址 关于可执行文件，有哪些 常见的格式呢，大家可以去查一下相关的资料 那么我们知道Windows操作系统下，可执行文件的格式呢是PE格式 在Linux操作系统下，可执行文件的格式呢是ELF格式 ELF，大家可以去自行了解一下这方面的知识 为了执行这个helloworld程序 操作系统首先要创建一个新的进程 并将helloworld程序的可执行文件格式 映射到该进程结构 表示由该进程来执行这个helloworld程序 做完了这件事情，操作系统就把控制权 交给了调度程序，我们假设调度程序正好选中了helloworld程序，那么 由操作系统为这个helloworld程序设置CPU 上下文环境，并跳到程序的开始之处 准备执行这条这个程序，那么下一个指令周期就是执行helloworld程序了 helloworld程序的执行的时候，当执行第一条指令的时候 会发生什么事情呢？实际上就是发生了 缺页异常，那么为什么呢？ 因为我们知道，程序在执行的时候，先要把程序的代码和数据读入内存 CPU才能去执行，但是

helloworld程序的代码和数据还没有读入内存呢 所以这个时候呢，硬件机制就会 捕获出缺页异常，并且把控制权交给操作系统 操作系统管理了计算机系统中的 内存，有很多的内存，比如说，我们说 有页式存储管理方案的话，那么内存就有很多很多的物理页面 操作系统的内存管理模块，就会分配一页空闲的物理内存 并且根据前面计算出的磁盘块地址 把helloworld程序的代码读入内存 从磁盘读入内存，然后 继续执行helloworld程序，有的时候程序很大 一页内存还不够，因此在这个执行过程中会多次地产生缺页异常 然后去到磁盘读入程序 到内存，这个过程会有多次 下面helloworld程序执行puts函数 那么puts函数的作用，是在标准的输出设备上 显示字符串，那么这个标准输出设备，通常就是我们的显示器或者说屏幕 那么puts函数呢，它实际上呢是一个系统调用 是由操作系统来完成这个功能 所以控制权又转回到操作系统 操作系统找到要将字符串送到哪一个显示设备 那么通常这个设备呢是由一个进程控制的，所以操作系统 把要写的字符串实际上是送给了这个进程 那么控制设备的进程，会告诉设备的窗口系统，它要显示字符串 窗口系统确定这是一个合法的操作，然后将字符串转换为像素 把像素写入设备的存储映像区 以下就是视频硬件的工作 视频硬件将像素，转换成显示器可以接收的一组控制信号或数据信号 那么显示器再去解释这个信号，激发液晶屏 其结果，OK，我们在 屏幕上就看到了字符串"hello world" 那么以上就是一个，非常简单的程序的一个执行过程 从启动到执行，到结束 从这个过程当中，我们得到什么结论呢？ 我们看到，在CPU上执行一个用户程序 这个用户程序会不时地去请求操作系统的服务 因此在CPU上，时而运行的是用户程序 时而呢，是操作系统程序在运行，我们来看一下 helloworld程序，在执行的时候呢，需要操作系统来创建进程 因此呢转到操作系统创建进程，进程创建完了以后呢，那么从操作系统返回 那么接着执行这个helloworld程序 那么出现了缺页异常以后呢，控制权又转回了操作系统 操作系统呢分配物理内存，给这个用户程序 然后呢又转向用户程序接着执行 然后helloworld程序又要去执行puts函数，又转向了操作系统 那么这是从用户程序，在执行 过程中，不断请求操作系统服务这个角度来看 程序执行的过程，如果我们从另外一个角度来看用户程序的执行呢？ 实际上呢，是操作系统 在执行过程中呢，如果选中了一个程序，那么 就去通过调度选中这个程序，去执行这个程序 程序执行过程中呢，会不断地去陷入操作系统 由操作系统完成一些服务，然后再通过调度再选中程序，接着执行 实际上是这

样一个过程，那么 从上面的介绍过程，我们看到在一个程序的 执行过程中，操作系统负责了这个程序的启动过程 负责了这个程序执行的过程，同时在执行的 过程中，不断地去为用户程序的执行 提供各种各样的支持