

第一章 引论

个人认为，学习知识时对其历史有一定了解是非常必要的，因为了解其历史和发展有助于我们理解它的一些操作和设定，所有关于操作系统的学习，我也将从引论开始学习。

操作系统

操作系统实际上就是一个直接管理计算机硬件的软件，大多数懂得编程的人并不完全了解硬件的运行方式，但是却有数万程序员可以编写出优秀的代码，这就是操作系统的功劳，它为用户程序提供了一个更好更简单清晰的计算机模型，并管理几乎所有的硬件设备。其最重要的作用之一是对系统的硬件资源进行调配。实际上操作系统的主要工作可以分为两类，分别是**资源管理**和**扩展机器**。操作系统是软件中最基础的部分，它运行在内核态。

内核态（也称管态、核心态）

在这个模式中操作系统具有对所有硬件的完全访问权，可以执行机器能够运行的任何指令。

用户态

出操作系统外的软件部分运行在用户态，在用户态下，只是用了机器指令中的一个子集。那些会影响机器的控制或可进行I/O操作的指令，在用户态中的冲虚里是禁止的。

CPU的指令模型

主要分为**流水线形式**和**超标量形式**。**流水线形式**顾名思义，这种形式主要分为三个单元，分别是**取指单元**、**解码单元**和**执行单元**，整个过程就像一个流水线一样工作，这样做的坏处是一旦一个指令进入流水线，那么它必须被执行完毕，即使这条指令是条件转移指令。**超标量形式**这种形式的好处是它有**多个取指单元**、**多个解码单元**和**多个执行单元**。它与流水线形式的最大差别在于它拥有一个**保持缓冲区**，这个缓冲区保存着解码完毕的指令，当某个执行单元空闲时就会去检查保持缓冲区，去执行缓冲区的任务，而这样做有一个缺点，就是执行的正确顺序可能会受到影响，这一部分的协调交给了硬件来解决。

储存器的类型

从小到大、从快到慢分别是 寄存器->高速缓存->主存->磁盘。高速缓存就是cpu的L1cache，L2cache等。其中主存就是我们通常说的计算机内存，也就是RAM。

几种不常见的操作系统类型 或者说常见但是不常听说的

传感器节点操作系统 从作者的言语中理解的这种操作系统类似于我们的手环，门窗开关检测，其主要作用就是供几个传感器使用，并且使用电池作为能源，对功耗要求较高，同时具有组网能力。**大型机操作系统** OS/390就是一种大型机操作系统，不过现在正在被Linux和UNIX的变体所取代。**智能卡操作系统** 万万没想到，原来信用卡里面用的是一个操作系统，甚至还跑了JAVA虚拟机。

硬实时系统和软实时系统的区别

硬实时系统必须时刻保证操作在规定时间内或时刻完成，而软实时系统虽然也希望保证操作在规定时间内或时刻完成，但软实时系统能够接受偶尔的失误。

管道Pipe

管道是一种虚文件，是用于进程间通信的一种方式，进程A若想与进程B通信，就需要二者提前设置一个管道，当A想向B发送数据时，进程A就将数据写到管道上，而B则可以通过读取这个管道得到数据。

特殊文件

操作系统中有个重要概念是特殊文件，它分为**块特殊文件**和**字符特殊文件**。这样做的目的是可以将I/O设备进一步抽象化，使各种的I/O设备使用起来的差距更小。**块特殊文件**指那些可随机存取的块组成设备，如磁盘等，比如打开一个块特殊文件，然后读第4块，程序可以直接访问设备的第4块而不必考虑存放该文件的文件系统结构。**字符特殊文件**用于打印机、调制解调器和其他接收或输出字符流的设备。按照惯例，特殊文件保存在/dev目录中。例如，/dev/lp是打印机（曾经称为行式打印机）。在Linux系统调试代码时经常会用到USB 或者一些串口，这些就是字符特殊文件，

操作系统结构

1. 单体系统：没看太明白，感觉就像是单片机上面运行的那种操作系统，就最简单的那种，它就是一整个系统一起运行，如果某个进程崩溃了，整个系统就会崩溃。
2. 层次式系统：也没看明白
3. 微内核：所谓微内核，就是只将非常必要的操作系统代码写入内核中，这样做的好处是操作系统内核只负责最主要的、直接与硬件接触或者危险的操作，其他的部分一概不写在内核中，如驱动程序等，这样做可以避免因为一个小错误而导致整个操作系统崩溃的情况，哪个部分崩溃就重启哪个部分即可。
4. 服务器-客户端：这种模式是微内核的一种略微变体，客户端和服务端通过消息传递的方式进行通信，客户端向服务器发送消息，使其提供服务，而实际上并不关注服务器和客户端是否运行在同一台机器或者不同机器上，因为服务器和客户端需要做的就是将消息发送给固定的客户端和接收固定对象的消息，并不关心消息的传递过程，ROS中就使用了这种模式。
5. 虚拟机：

再生服务器(reincarnation server)

其任务是检查其他服务器和驱动器的功能是否正确。一旦检查出一个错误，它自动取代之，无须任何用户的干预。这种方式使得系统具有自修复能力，并且获得了较高的可靠性。

系统对每个进程的权限有着许多限制。正如已经提及的，**设备驱动器只能与授权的I/O端口接触**，对内核调用的访问也是按单个进程进行控制的，这是考虑到进程具有向其他多个进程发送消息的能力。

shell和GUI都是用户接口程序

shell 基于文本的人机交互程序

GUI(Graphical User Interface)图形用户界面 基于图形的人机交互程序

POSIX 所有UNIX 系统都必须支持的小型系统调用接口。