

# 【OS】Day4

|         |                    |
|---------|--------------------|
| ▼ Class | 现代操作系统             |
| 📅 Date  | @December 10, 2021 |

## 【Ch1】引论

### 1.5 操作系统概念

#### 1.5.1 进程

进程(process)本质上是正在执行的一个程序。与每个进程相关的是地址空间(address space)，这是从某个最小值的存储位置（通常是零）到某个最大值的存储位置的列表。

在这个空间地址中，进程可以进行读写。该地址空间中存放有可执行程序、程序的数据以及程序的堆栈。

进程基本上是容纳运行一个程序所需要所有信息的容器。

一个进程被暂时挂起后，在随后的某个时刻里，该进程再次启动时的状态必须与先前暂停时完全相同。这就意味着在挂起时该进程的所有信息都要保存下来。

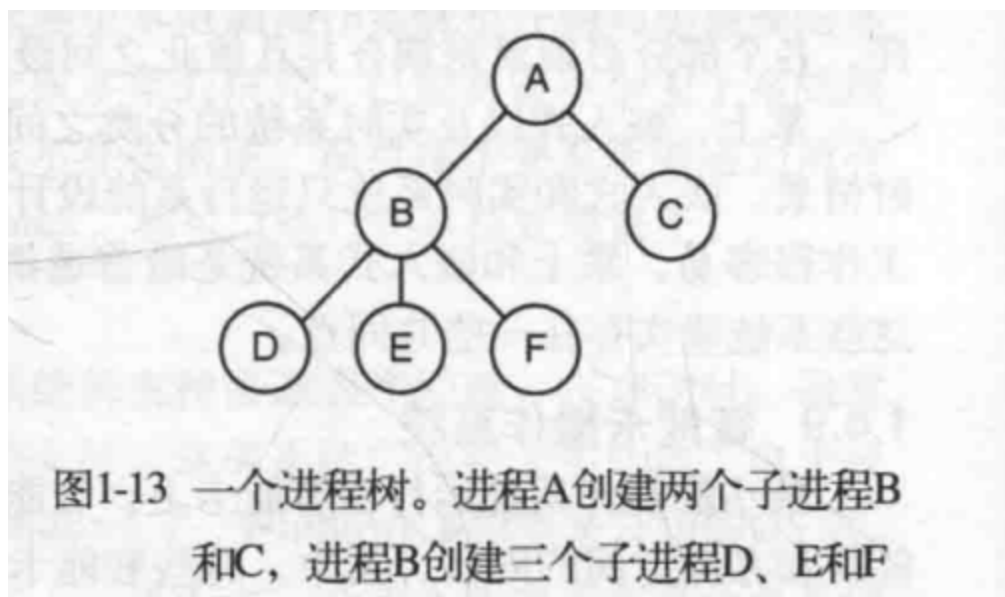
在许多操作系统中，与一个进程有关的所有信息，除了该进程自身地址空间的内容以外，均存放在操作系统的一张表中，称为进程表(process table)，进程表是数组（或链表）结构，当前存在的每个进程都要占用其中一项

所以，一个（挂起的）进程包括：进程的地址空间（往往称作磁芯映像, core image），以及对应的进程表项（其中包括寄存器以及稍后重新启动该进程所需要的许多其他信息）

与进程管理有关的最关键的系统调用是那些进行进程创建和进程终止的系统调用。

考虑一个典型的例子：一个称为命令解释器(command interpreter)或shell的进程从终端上读命令。此时，用户刚键入一条命令要求编译一个程序。shell必须先创建一个新进程来执行编译程序。当执行编译的进程结束时，它执行一个系统调用来终止子集

若一个进程能够**创建一个或多个进程**（称为**子进程**），而且这些进程又可以创建子进程，则很容易得到**进程树**



合作完成某些作业的相关进程经常需要**彼此通信**以便**同步它们的行为**。这种通信称为**进程间通信(interprocess communication)**

其他可用的进程系统调用包括：**申请更多的内存（或释放不再需要的内存）、等待一个子进程结束、用另一个程序覆盖该程序等**

系统管理器授权每个进程使用一个给定的**UID(User Identification)**。每个被启动的进程都有一个启动该进程的用户UID。**子进程与父进程有一样的UID**

### 1.5.2 地址空间

**如果一个进程有比计算机拥有的主存还大的地址空间，而且该进程希望使用全部的内存，该如何实现？**

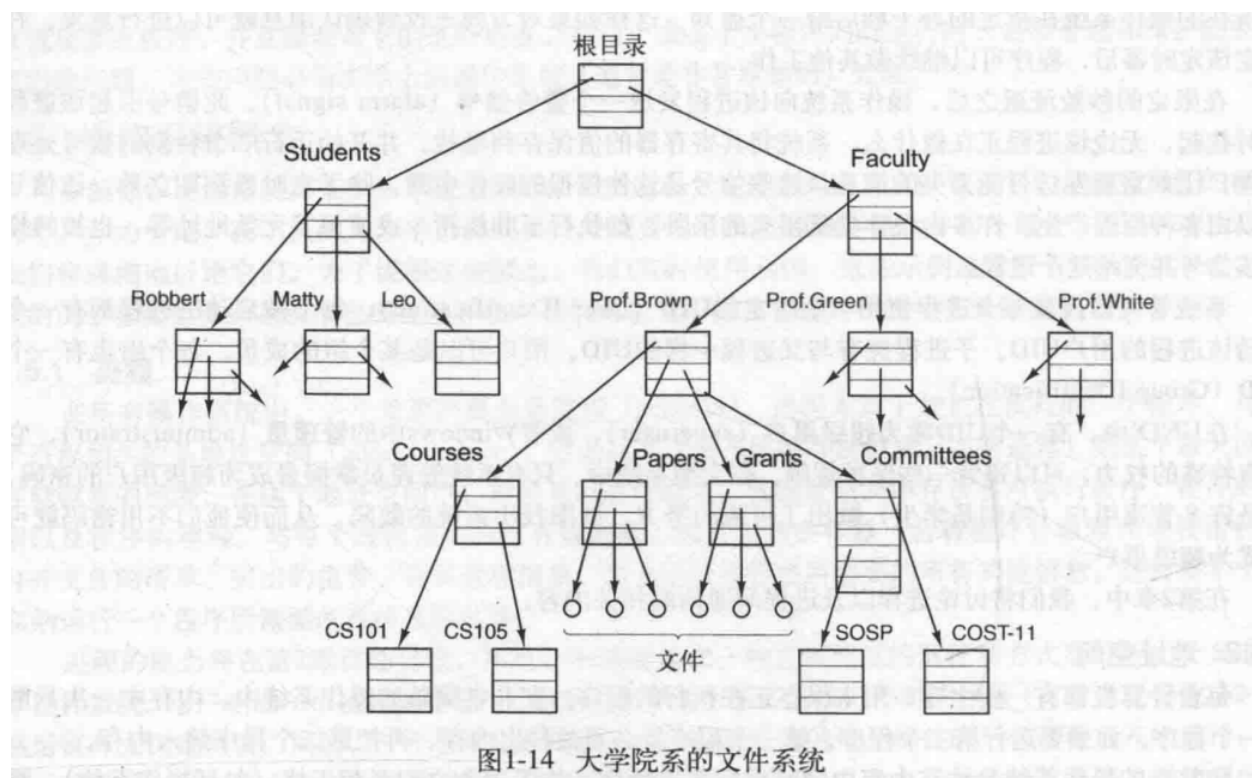
在现代操作系统中，有了一种称为**虚拟内存的技术**。操作系统可以把**部分地址空间装入主存，部分留在磁盘上**，并在需要时来回交换他们。

操作系统创建了一个地址空间的抽象，作为进程可以引用地址的集合

### 1.5.3 文件

支持操作系统的另一个关键概念是文件系统。操作系统的一项主要功能是隐藏磁盘和其他I/O设备的细节特性，提供程序员一个良好的独立于设备的抽象文件模型

为了保存文件的地方，大多数操作系统支持目录(directory)的概念，从而可把文件分类成组。目录项可以是文件或者目录，这样就产生了层次结构：文件系统



进程和文件层次都可以组织成树状结构，但这两种树状结构有不少的不同之处：

- 一般进程的树状结构层次不深（很少超过三层）；而文件树状结构的层次常常多达四层、五层甚至更多
- 进程树状结构是暂时的，通常最多存在几分钟；而目录层次则可能存在数年之久
- 进程和文件在所有权及保护方面也是有区别的。典型的，只有父进程能够控制和访问子进程；而在文件和目录中通常存在一种机制，使文件所有者之外的其他用户也可以访问该文件

目录层结构中的所有文件都可以通过从目录的顶部即根目录(root directory)开始的路径名(path name)来确定。

绝对路径名包括了从根目录到该文件的所有目录清单，它们之间用正斜线 / 隔开。如图，文件CS101的路径名是 /Faculty/Prof.Brown/Courses/CS101。最开始的正斜线表示这是从根

目录开始的绝对路径。

处于历史原因，在Windows中用反斜杠 \ 作为分隔符

在实例中，每个进程有一个工作目录(working directory)。对于没有以斜线开头给出绝对地址的路径，将在这个工作目录下寻找。如果/Faculty/Prof.Brown是工作目录，Course/CS101与上面给定的绝对路径名表示的是同一个文件。

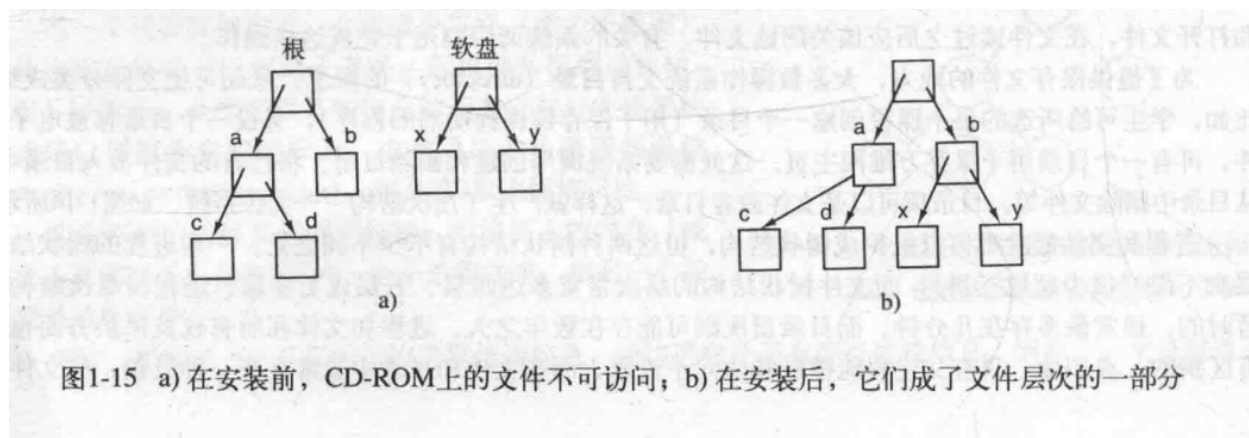
进程可以通过使用系统调用指定新的工作目录，从而改变工作目录

在读写文件之前，首先要打开文件，检查其访问权限。

若权限许可，系统将返回一个小整数，称作文件描述符(File Descriptor)，供后续操作使用。若禁止访问，系统则返回一个错误码。

UNIX中的另一个重要概念是安装文件系统。大多数台式机都有一个或多个光盘驱动器，可以插入CD-ROM、DVD和蓝光光盘。为了提供一个出色的方式处理可移动介质，UNIX允许把光盘上的文件系统接到主文件树上。

在 mount 调用之前，根文件系统在硬盘上，而第二个文件系统在CD-ROM上，它们是分离且无关的



然而，不能使用CD-ROM上的文件系统，因为上面没有可指定的路径。UNIX不允许在路径前面加上驱动器名称或代码

mount 系统调用允许把在CD-ROM上的文件系统连接到程序所希望的根文件系统上。

在UNIX中，另一个重要的概念是特殊文件(special file)。提供特殊文件是为了使I/O设备看起来像文件一般。这样，就像使用系统调用读写文件一样，I/O设备也可以通过同样的系统调用进行读写。

管道(pipe)是一种虚文件，它可以连接两个进程。如果进程A希望和进程B通过管道对话，它们必须提前设置该管道。当进程A相对进程B发送数据时，它把数据写到管道上，仿佛管道就是输出文件一样