

操作系统作业1

罗兴攀 PB19051150

1.系统功能

系统角度：操作系统是一个控制程序，可以控制程序的执行，以防止对计算机的错误使用；操作系统还是资源分配器，管理计算机资源，决定在冲突的请求之间公平且有效地分配资源。

用户角度：操作系统对不同的用户提供的功能（服务）不尽相同。对PC用户想要方便，易用以及优秀的性能表现，而并不关心资源利用率；但共享式的计算机，比如大型机和小型机，就必须让所有的用户都满意，即操作系统需要尽可能提高资源利用率；使用专用系统如工作站的用户有自己的专用资源，但也频繁的使用来自服务器的共享资源，此时操作系统需要保证公平；可移动设备资源较为匮乏，操作系统着重优化了可用性以及持久性；一些微型计算机甚至没有用户界面，此时操作系统只在后台运行，这样的操作系统往往比较重视安全和稳定，而很少提供与用户的交互。

2.Multiprogramming & Multitasking

Multiprogramming: 多程序编程出于提高效率的需要，一个任务可能并不需要一直占用CPU，比如它可能需要等待I/O的完成；因此，为了使CPU始终处于工作（忙碌）状态，将所有作业存储在作业池中，通过工作调度，使CPU始终处于工作状态，从而提高效率。

Multitasking: 多任务处理又称时间共享，实质就是通过频繁切换作业，使用户可以在每个作业运行时都与其进行交互。多任务处理的目的，一是保证公平，每个用户或作业都可以几乎随时使用CPU资源，不必等待很久，其次是由于工作切换较快，可以大大提高用户与运行中的程序进行交互的体验，当然多任务处理也允许多个用户共享计算机资源。

3.存储的层次和缓存

储存的层次结构和缓存：对我们用户而言，我们感觉多个任务是在同时进行的，这是操作系统通过Multitasking造成的错觉，事实上，由于CPU核的数目有限，只有部分进程“真正的同时运行”，而其它的进程（包括其数据）需要先存储在其它地方。而不同的存储设备读写性能不同，成本也大有差距；程序越“靠近”CPU，即离真正运行越近，就需要位于更高级的存储设备上，这就造成了存储的层次结构。进程可能首先位于硬盘上，然后被拷贝一份到内

存中，之后又被拷贝到高速缓存上，最后才能到达CPU的硬件寄存上实现运行。缓存，又称高速缓存技术就是上述描述的过程，进程信息首先保存在某个存储系统中，需要使用使，会被临时复制到更快的存储系统中。

4.系统调用及API

系统调用和函数调用类似，但不同的是，系统调用的实现实在操作系统的内部实现的；系统调用作为进程和操作系统内核之间的编程接口，为用户程序提供了要求操作系统执行任务的方法。

一个简单的程序可能就需要大量的系统调用，对用户十分不友好，因此，将系统调用封装为更易用的程序接口就是API. API即应用程序接口，是可供程序员使用的一组函数。API使用系统调用以及高级编程语言，得到一些常用的函数（方法），简化了用户程序编写，用户可以通过API间接进入系统内核，比如在内核外部使用open()函数，而open()函数的内部实现就调用了系统调用，访问或使用了系统资源，通过API，用户可以不必要知道API的具体实现，也不需要知道它在执行期间做了什么，就可以更方便的使用系统资源。

5.Dual Mode

Dual Mode 即双模式，为了确保操作系统的正确运行，必须区分操作系统代码和用户代码的执行；操作系统具有用户模式和内核模式（监视模式、系统模式）两种运行模式，由硬件提供的 mode bit提供了区分系统是在运行用户代码还是内核代码的能力，系统调用会将模式改为内核，调用完成后的返回将模式重置为用户。

双模式的目的是允许操作系统保护自己以及其它一些系统组件。

6.操作系统提供的服务

- 执行程序：加载程序，运行和结束程序，指示异常等
- I/O操作：提供用户进行I/O的方法等
- 文件系统操作：读写文件和目录，创建、删除文件和目录，搜索，列表，权限管理等
- 通讯服务：进程间的信息交换，包括共享内存、消息传递等
- 错误检测：检测CPU、内存硬件、I/O设备及用户程序等的错误，并进行合理的错误处理
- 资源分配：将有限的计算机资源分配给不同的作业、不同的用户
- 用户界面：提供命令行、批处理文件、图形用户界面等
- 保护和安全

7.各种操作系统结构的优缺点

Monolithic 单片结构

所有功能组件在一个层次上，优点是性能良好，系统调用接口和内核通信的开销非常小，但随着功能的增加与扩展，难以设计与维护。

分层方法（层次化结构）

主要优点在于简化了构造和调试，所选的层次要求每层只能调用更低层的功能和服务，随着层次升高，DS、Ops及硬件得以隐藏。缺点是难以合理定义各层，且由于实现需要层层递进，执行效率较低。

微内核结构

为内核的主要功能是，为客户端程序和运行在用户空间中的各种服务提供通信；这种结构的优点是便于扩展操作系统，因为微内核本身就很很小，大部分新服务可在用户空间内增加，而不需要或仅需要微量修改内核。微内核也更容易移植到新的架构或硬件平台上，同时，由于内核模式下运行的代码更少，系统也就更可靠、更安全。

缺点则是增大了用户空间到内核空间的性能开销

模块化结构

模块化结构提供了一套核心组件，并允许通过模块链接来附加服务；类似于分层系统，但任何模块都可以调用其它模块，因此比分层结构更加灵活；也类似于微内核，主内核只提供核心功能，但模块化结构无需调用消息传递，更加高效。

模块化结构综合了分层结构和微内核的优点，大多数现代操作系统都实现了可加载内核模块，无明显缺点。

8.机制与策略分离

机制与策略分离就是将“如何做”和“做什么”分离开来实现。例如，定时器是保护CPU的机制，那么定时器“如何做”才能保护CPU就是机制，需要将定时器的功能进行单独的实现，而定时器的机制确定后，我们需要使用定时器来“做什么”就是策略问题，我们希望的是，使用定时器在不同场合时都尽可能不需要修改定时器原先的实现，当为不同的用户在不同的场景

下使用定时器时，我们不需要考虑改变定时器的内部实现，这种对外部策略不敏感的通用机制是更加合理的。

策略与机制分离对系统设计的灵活性至关重要，策略是灵活多变的，倘若每次策略的改变都会导致底层机制的改变，将会给系统设计带来极大的不便。