2-1

操作系统提供的一类服务是在系统中同时运行的不同进程之间实施保护。 允许进程仅访问与其地址空间关联的那些内存位置。 此外，不允许进程破坏与其他用户关联的文件。 在没有操作系统干预的情况下，也不允许进程直接访问设备。 操作系统提供的第二类服务是提供底层硬件不直接支持的新功能。 虚拟内存和文件系统是操作系统提供的两种新服务示例。

2-2

1. 传递寄存器中的参数。
2. 寄存器传递参数块的起始地址。
3. 参数可以由程序压进到堆栈上，并由操作系统弹出堆栈。

2-3

可以发出周期性的定时器中断，并监视在传递中断时当前正在执行的指令或代码段。 哪些代码片段处于活动状态的统计配置文件应该与程序在其代码的不同部分花费的时间一致。 一旦获得了这样的统计配置文件，程序员就可以优化消耗更多CPU资源的那些代码段。

2-4

1. 文件的创建和删除。
2. 目录的创建和删除。
3. 支持原语操作文件和目录。
4. 将文件映射到辅助存储。
5. 在非易失性存储介质上备份文件。

2-5  
可以访问每个设备，就好像它是文件系统中的文件一样。 由于大多数内核通过此文件接口处理设备，因此通过实现特定于硬件的代码来支持此抽象文件接口，可以相对轻松地添加新的设备驱动程序。 因此，这有利于开发用户程序代码和设备驱动程序代码，用户程序代码可以以相同的方式写入访问设备和文件，设备驱动程序代码可以编写以支持定义良好的API。 使用相同接口的缺点在于，可能难以在文件访问API的上下文中捕获某些设备的功能，从而导致功能丧失或性能损失。 其中一些可以通过使用ioctl操作来克服，该操作为进程调用设备上的操作提供了通用接口。

2-6

用户应该能够使用操作系统提供的系统调用接口开发新的命令解释器。 命令解释器允许用户创建和管理进程，并确定它们进行通信的方式（例如通过管道和文件）。 由于所有这些功能都可以由用户级程序使用系统调用来访问，因此用户应该可以开发新的命令行解释器。

2-7

进程间通信的两种模型是消息传递模型和共享内存模型。 消息传递对于交换较少量的数据很有用，因为不需要避免冲突。 与用于计算机间通信的共享内存相比，它更容易实现。 共享内存允许最大的通信速度和便利性，因为它可以在计算机内进行时以内存传输速度完成。 但是，此方法会损害共享内存的进程之间的保护和同步。

2-8