

操作系统第一次作业

BobAnkh

March 2021

1. CPU、内存、磁盘、网卡、打印机、键盘以及显示器，这些资源能够采用哪种复用形式，时间复用、空间复用或者两者皆可？

答：采用列表展示如下，在各资源能够采用的复用形式中打√：

资源 \ 复用形式	时间复用	空间复用
CPU	√	
内存		√
磁盘		√
网卡	√	
打印机	√	
键盘	√	
显示器	√	√

2. 什么是陷入指令？说明其用途。

答：陷阱（Trap）是能够使系统从用户态转换到核心态的指令，以执行只有在核心态下才能执行的指令，其主要用途是实现系统调用——系统调用是操作系统提供给编程人员的唯一接口；利用系统调用，可以动态请求和释放系统资源，完成与硬件相关的工作以及控制程序的执行等。

3. 系统调用和普通的过程调用（函数调用）有什么区别？

答：两者的主要区别如下：

- 系统调用在内核态实现，是操作系统的一个入口点，为应用程序提供了必要的访问硬件设备的接口，而普通过程调用在用户态实现，与用户程序相联系，不直接触及硬件设备的访问；
- 系统调用需要借助陷入指令实现，而普通过程调用不需要；
- 系统调用由内核函数实现，而普通过程调用由一般函数实现；
- 系统调用需要在用户空间和内核上下文环境间切换，开销较大，而普通过程调用开销较小；
- 系统调用在内核地址空间执行，普通过程调用在用户地址空间执行；

4. 操作系统一般要对系统调用进行封装，以库过程调用或者 API 的形式提供给用户，为什么？

答：其一是为了提高用户程序的可移植性——系统调用是依赖操作系统的，在不同平台上的实现不同，而封装成库过程调用则可获得较好的平台移植性；其二是为了程序逻辑实现的简便，使得用户编程更加方便快捷，代码更简洁明了。

5. 在程序设计的过程中，应该尽可能多的采用系统调用还是相反，给出你的理由。

答：两种策略各有利弊，所以应该在不同的时候采用不同的策略。多采用系统调用可以使得编程更加简洁，代码更易读，但是频繁从用户态切换到内核态的开销以及多核下系统调用可能导致的 CPU 切换进程，都会导致一定的性能下降，在通常对于性能没有过高追求的情况下可以考虑多使用系统调用；而对于一些对性能要求非常高的程序来说，它们虽然希望利用系统调用的服务，但却希望能够尽量避免陷入/返回和系统调用处理程序带来的开销，尽可能提高性能，因此采用由内核直接调用系统调用服务例程的方式，例如 httpd 就是这样做的——它为了避免上述开销，直接从内核调用 socket 等系统调用服务例程，当然这样较少利用系统调用也是以编程较为繁琐、代码易读性不好作为代价的。

6. 一台计算机具有单一的 CPU 和两台输入/输出设备 DEV1、DEV2，其中 CPU、DEV1、DEV2 都能并行工作。现有 3 个作业同时投入运行，执行顺序为 Job1、Job2、Job3，这三个作业对 CPU 和输入/输出设备的使用顺序和时间如下：

Job1: DEV2(30ms)、CPU(10ms)、DEV1(30ms)、CPU(10ms)、DEV2(20ms)

Job2: DEV1(20ms)、CPU(20ms)、DEV2(40ms)

Job3: CPU(30ms)、DEV1(20ms)、CPU(10ms)、DEV1(10ms)

若采用单道批处理操作系统，计算各个作业的周转时间以及 CPU、DEV1 和 DEV2 的利用率。若采用多道批处理操作系统，计算各个作业的周转时间以及 CPU、DEV1 和 DEV2 的利用率。

答：

• 采用单道批处理操作系统

- Job1 周转时间：30 + 10 + 30 + 10 + 20 = 100ms
- Job2 周转时间：100 + 20 + 20 + 40 = 180ms
- Job3 周转时间：180 + 30 + 20 + 10 + 10 = 250ms
- CPU 利用率： $\frac{10+10+20+30+10}{250} = 32\%$
- DEV1 利用率： $\frac{30+20+20+10}{250} = 32\%$
- DEV2 利用率： $\frac{30+20+40}{250} = 36\%$

• 采用多道批处理操作系统

将流程绘制如下图 (其中每一格代表 10ms, 黄色表示使用 CPU, 绿色表示使用 DEV1, 蓝色表示使用 DEV2, 白色表示等待):

Job1												
Job2												
Job3												



由上图得到各周转时间和利用率：

- Job1 周转时间：120ms
- Job2 周转时间：100ms
- Job3 周转时间：90ms
- CPU 利用率： $\frac{80}{120} = 66.7\%$
- DEV1 利用率： $\frac{80}{120} = 66.7\%$
- DEV2 利用率： $\frac{9}{120} = 75\%$