

1. Describe the differences among short-term, medium-term, and long-term scheduling.

长期调度：发生频率较低，控制着程序进入系统竞争 CPU，控制并发展，平衡 I/O 和 CPU 资源。

中期调度：发生频率中等，将进程在内-外存间交换，以管理内存负载，缓解内存压力，实现交换。

短期调度：发生频率非常高（毫秒级），决定就绪状态的哪个进程获得 CPU，提高 CPU 利用率，优化响应时间。

2. Describe the actions taken by a kernel to context-switch between processes.

一般来说，作业系统必须保存当前运行的进程的状态并恢复预定要运行的进程的状态。保存进程的状态通常包括所有 CPU 寄存器的值，以及内存分配。上下文切换必须执行许多特定于架构的操作，包括清除数据和指令缓存。

3. 采用下述程序，确定 A、B、C、D 四行中 pid 和 pid1 的值。（假设父进程和

子进程的 pid 分别为 2600 和 2603）

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
    pid_t pid, pid1;
    pid=fork();
    if (pid<0)
    {
        fprintf(stderr,"fork fail!");
        return 1;
    }
    else if (pid==0)
    {
        pid1=getpid();
        printf("child:pid=%d",pid);      //A
        printf("child:pid1=%d",pid1);    //B
    }
    else
    {
        pid1=getpid();
        printf("parent:pid=%d",pid);    //C
        printf("parent:pid1=%d",pid1);  //D
        wait(NULL);
    }
    return 0;
}
```

A: pid = 0      B: pid1 = 2603

C: pid = 2603    D: pid1 = 2600