

1日上午12时UTC (Universal Time Coordinated, 协调世界时, 以前称为格林威治平均时) 以来的时钟滴答数, 或者转换成自某个其他标准时间以来的时钟滴答数。Windows的时间原点是1980年1月1日。每一次时钟滴答都使实际时间增加一个计数。通常会提供实用程序来手工设置系统时钟和备份时钟, 并且使两个时钟保持同步。

5.5.2 时钟软件

时钟硬件所做的全部工作是根据已知的时间间隔产生中断。涉及时间的其他所有工作都必须由软件——时钟驱动程序完成。时钟驱动程序的确切任务因操作系统而异, 但通常包括下面的大多数任务:

- 1) 维护日时间。
- 2) 防止进程超时运行。
- 3) 对CPU的使用情况记账。
- 4) 处理用户进程提出的alarm系统调用。
- 5) 为系统本身的各个部分提供监视定时器。
- 6) 完成概要剖析、监视和统计信息收集。

时钟的第一个功能是维持正确的日时间, 也称为实际时间 (real time), 这并不难实现, 只需要如前面提到的那样在每个时钟滴答将计数器加1即可。惟一要当心的事情是日时间计数器的位数, 对于一个频率为60Hz的时钟来说, 32位的计数器仅仅超过2年就会溢出。很显然, 系统不可能在32位中按照自1970年1月1日以来的时钟滴答数来保存实际时间。

可以采取三种方法来解决这一问题。第一种方法是使用一个64位的计数器, 但这样做使维护计数器的代价很高, 因为1秒内需要做很多次维护计数器的工作。第二种方法是以秒为单位维护日时间, 而不是以时钟滴答为单位, 该方法使用一个辅助计数器来对时钟滴答计数, 直到累计完整的一秒。因为 2^{32} 秒超过了136年, 所以该方法可以工作到22世纪。

第三种方法是对时钟滴答计数, 但是这一计数工作是相对于系统引导的时间, 而不是相对于一个固定的外部时间。当读入备份时钟或者用户输入实际时间时, 系统引导时间就从当前日时间开始计算, 并且以任何方便的形式存放在内存中。以后, 当请求日时间时, 存储的日时间值加到计数器上就可以得到当前的日时间。所有这三种方法如图5-33所示。

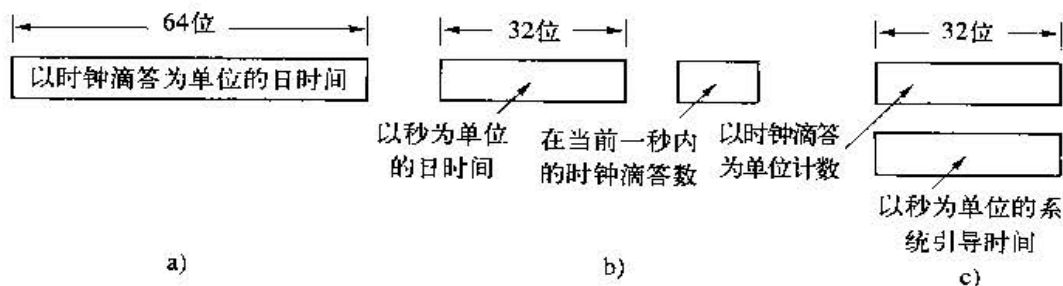


图5-33 维护日时间的三种方法

时钟的第二个功能是防止进程超时运行。每当启动一个进程时, 调度程序就将一个计数器初始化为以时钟滴答为单位的该进程时间片的取值。每次时钟中断时, 时钟驱动程序将时间片计数器减1。当计数器变为0时, 时钟驱动程序调用调度程序以激活另一个进程。

时钟的第三个功能是CPU记账。最精确的记账方法是, 每当一个进程启动时, 便启动一个不同于主系统定时器的辅助定时器。当进程终止时, 读出这个定时器的值就可以知道该进程运行了多长时间。为了正确地记账, 当中断发生时应该将辅助定时器保存起来, 中断结束后再将其恢复。

一个不太精确但更加简单的记账方法是在一个全局变量中维护一个指针, 该指针指向进程表中当前运行的进程的表项。在每一个时钟滴答, 使当前进程的表项中的一个域加1。通过这一方法, 每个时钟滴答由在该滴答时刻运行的进程“付费”。这一策略的一个小问题是: 如果在一个进程运行过程中多次发生中断, 即使该进程没有做多少工作, 它仍然要为整个滴答付费。由于在中断期间恰当地对CPU进行记账的方法代价过于昂贵, 因此很少使用。