

Xlib 中定时器的使用

BG6RDF

Xlib 是 X Window 系统最底层的程序设计接口。尽管现在 X Window 中已有很多工具集和 Widget 提供了高层接口，方便了程序设计，但有时为了灵活性，还需要使用 Xlib。

Unix 中设置进程定时器通常方法是：编写信号处理程序(signal handler)；调用 `setitimer` 打开定时器，安装信号处理程序。操作系统会在定时器到期后调用信号处理程序。

有时需要在信号处理程序中调用 Xlib 函数，例如在定时器到期后打开一个窗口。但是信号处理程序中是不能调用 Xlib 函数的。原因是 Xlib 函数是不可重入的，如果在 Xlib 函数执行时，触发了定时器信号处理程序，该处理程序调用另外一个 Xlib 函数，会引起混乱。

这个问题的解决办法有两种，下面一一加以介绍：

1. 在定时器信号处理程序中设置标志变量，在主事件循环中判断该变量是否改变。通常 Xlib 的事件循环都使用 `XNextEvent` 函数，这个函数在未接收到事件前是不会返回的，即该函数是阻塞 (blocking) 函数。因此只有在接收到事件后，主事件循环才可能对标志变量进行判断。要克服这个问题可以使用 `XtAppNextEvent` 函数，但该函数不是 Xlib 函数，本文不作讨论。
2. 要用纯 Xlib 的方法解决这个问题，必须满足这样一些条件：
`XNextEvent` 不阻塞；事件循环中不得对事件进行轮巡，否则会对系统性能造成影响；必须能及时发现事件并调用相应的事件处理程序。为使 `XNextEvent` 不阻塞可使用 `Xpending` 函数先判断是否有事件需要处理，如果有再调用 `XNextEvent`，此时不会阻塞。为了避免轮巡并能及时发现事件，可使用 Unix 系统中著名的 `select` 函数。`Select` 函数可对指定的文件描述符上的 I/O 进行等待(阻塞)，并且可以指定等待的时间。但是对应的文件描述符是什么呢？答案就是 Xlib 中的一个宏 `ConnectionNumber`，该宏返回与 X Server 连接的一个文件描述符。这种方法的整个事件处理循环流程片段是：

```
while (!ShutDown) { //ShutDown 标志变量用以判断是否退出消息循环
    if (XPending(Disp)) {
        XEvent e;
        XNextEvent(Disp, &e);
        EventHandler(&e); //进行该事件相应的
    } else {
        //display 是由 XopenDisplay 返回的 Display *
        int xfd = ConnectionNumber(display);
        fd_set rfd;
        OBJ_LIST_NODE * p;
        timeval now, tm;
        FD_ZERO(&rfd);
        FD_SET(xfd, &rfd);
        gettimeofday(&now, 0);
```

```

//LastTime 是 timeval 类型的变量，保存上次时间
//ticks 的单位是毫秒 millisecond
long ticks=(now.tv_sec-LastTime.tv_sec)*1000+
    (now.tv_usec-LastTime.tv_usec)/1000;
//TimerList 是一个定时器事件列表
for(p=App->TimerList.GetHeaderNode();
    p!=0;
    p=App->TimerList.GetNextNode(p)) {
//TimerTick 是定时器事件的函数用以判断是否到期
//如果到期则调用有关函数
    p->TimerTick(ticks);
}
gettimeofday(&LastTime, 0);
//select 等待 100 毫秒
tm.tv_sec=0l;
tm.tv_usec=100000l;
    select(xfd + 1, &rfd, 0, 0, &tm);
} /* End else*/
} /* End while */

```

该程序在 Redhat Linux 6.2 下调试通过。该段程序在所附源代码的 Application.cc 文件中，编译方法是 make xwintest1。

但是第二种方法也有一定缺点：因为该事件循环是对事件优先处理的，所以不适合于对定时器精度要求较高的应用。

参考资料：

1. <http://www.faqs.org/>
2. <http://www.rahul.net/kenton/xglossary.html>
3. 《Advanced Programming in the UNIX Environment》 W.Richard Stevens