**event\_base\_loop启动event\_base**

一旦有了一个已经注册了某些事件的 event\_base, 就需要让 libevent 等待事件并且通知事件的发生。

运行event\_base函数如下

#define EVLOOP\_ONCE 0x01

#define EVLOOP\_NONBLOCK 0x02

#define EVLOOP\_NO\_EXIT\_ON\_EMPTY 0x04

int event\_base\_loop(struct event\_base \*base, int flags);

默认情况下,event\_base\_loop()函数运行 event\_base 直到其中没有已经注册的事件为止

EVLOOP\_ONCE：循环将等待某些事件成为激活的（可读或可写等）, 执行激活的事件的回调函数，然后返回

EVLOOP\_NONBLOCK：循环不会等待事件被触发，循环将仅仅检测是否有事件已经就绪，如果有，则执行事件的回调，然后返回，没有则直接返回

**event\_base\_loop处理过程**

while (any events are registered with the loop,

or EVLOOP\_NO\_EXIT\_ON\_EMPTY was set) {

if (EVLOOP\_NONBLOCK was set, or any events are already active)

If any registered events have triggered, mark them active.

else

Wait until at least one event has triggered, and mark it active.

for (p = 0; p < n\_priorities; ++p) {

if (any event with priority of p is active) {

Run all active events with priority of p.

break; /\* Do not run any events of a less important priority \*/

}

}

if (EVLOOP\_ONCE was set or EVLOOP\_NONBLOCK was set)

break;

}

**event\_base\_dispatch**

int event\_base\_dispatch(struct event\_base \*base);

event\_base\_dispatch ()等同于没有设置标志的 event\_base\_loop ( )