```
#include <linux/timer.h
```

- define a timer

> struct timer_list wd_timer;

- initialize the timer

> init_timer(&wd_timer);

- config the timer

```
#define setup_timer(timer, fn, data)                    \
    __setup_timer((timer), (fn), (data), 0)
```

> setup_timer(&wd_timer, wd_timer_expire, 0);

```
static void wd_timer_expire(unsigned long data)
{
}
```

wd_timer_expire()是time out后被调用的callback.

- delete the timer

> del_timer_sync(&wd_timer);

- modify a timer's timeout

```
/**
 * mod_timer - modify a timer's timeout
 * @timer: the timer to be modified
 * @expires: new timeout in jiffies
 *
 * mod_timer() is a more efficient way to update the expire field of an
 * active timer (if the timer is inactive it will be activated)
 *
 * mod_timer(timer, expires) is equivalent to:
 *
 *     del_timer(timer); timer->expires = expires; add_timer(timer);
 *
 * Note that if there are multiple unserialized concurrent users of the
 * same timer, then mod_timer() is the only safe way to modify the timeout,
 * since add_timer() cannot modify an already running timer.
 *
 * The function returns whether it has modified a pending timer or not.
 * (ie. mod_timer() of an inactive timer returns 0, mod_timer() of an
 * active timer returns 1.)
 */
int mod_timer(struct timer_list *timer, unsigned long expires)
```

```
mod_timer(&wd_timer, jiffies + msecs_to_jiffies(wd_service_interval));
```

这里wd_service_interval是timeout值，毫秒，millisecond。