timer_stats - timer usage statistics

timer_stats is a debugging facility to make the timer (ab)usage in a Linux system visible to kernel and userspace developers. If enabled in the config but not used it has almost zero runtime overhead, and a relatively small data structure overhead. Even if collection is enabled runtime all the locking is per-CPU and lookup is hashed.

timer_stats should be used by kernel and userspace developers to verify that their code does not make unduly use of timers. This helps to avoid unnecessary wakeups, which should be avoided to optimize power consumption.

It can be enabled by CONFIG_TIMER_STATS in the "Kernel hacking" configuration section.

timer_stats collects information about the timer events which are fired in a Linux system over a sample period:

- the pid of the task(process) which initialized the timer
- the name of the process which initialized the timer
- the function where the timer was intialized
- the callback function which is associated to the timer
- the number of events (callbacks)

timer_stats adds an entry to /proc: /proc/timer_stats

This entry is used to control the statistics functionality and to read out the sampled information.

The timer_stats functionality is inactive on bootup.

To activate a sample period issue:
```
# echo 1 >/proc/timer_stats
```

To stop a sample period issue:
```
# echo 0 >/proc/timer_stats
```

The statistics can be retrieved by:
```
# cat /proc/timer_stats
```

The readout of /proc/timer_stats automatically disables sampling. The sampled information is kept until a new sample period is started. This allows multiple readouts.

Sample output of /proc/timer_stats:

```
Timerstats sample period: 3.888770 s
  12,     0 swapper          hrtimer_stop_sched_tick (hrtimer_sched_tick)
  15,     1 swapper          hcd_submit_urb (rh_timer_func)
   4,   959 kedac            schedule_timeout (process_timeout)
   1,     0 swapper          page_writeback_init (wb_timer_fn)
  28,     0 swapper          hrtimer_stop_sched_tick (hrtimer_sched_tick)
  22,  2948 IRQ 4            tty_flip_buffer_push (delayed_work_timer_fn)
   3,  3100 bash             schedule_timeout (process_timeout)
   1,     1 swapper          queue_delayed_work_on (delayed_work_timer_fn)
```

```
   1,     1 swapper          queue_delayed_work_on (delayed_work_timer_fn)
   1,     1 swapper          neigh_table_init_no_netlink (neigh_periodic_timer)
   1,  2292 ip               __netdev_watchdog_up (dev_watchdog)
   1,    23 events/1         do_cache_clean (delayed_work_timer_fn)
90 total events, 30.0 events/sec
```

The first column is the number of events, the second column the pid, the third column is the name of the process. The forth column shows the function which initialized the timer and in parenthesis the callback function which was executed on expiry.

Thomas, Ingo

Added flag to indicate 'deferrable timer' in /proc/timer_stats. A deferrable timer will appear as follows

```
  10D,     1 swapper          queue_delayed_work_on (delayed_work_timer_fn)
```