

## IRQ-flags state tracing

started by Ingo Molnar <mingo@redhat.com>

the "irq-flags tracing" feature "traces" hardirq and softirq state, in that it gives interested subsystems an opportunity to be notified of every hardirqs-off/hardirqs-on, softirqs-off/softirqs-on event that happens in the kernel.

CONFIG\_TRACE\_IRQFLAGS\_SUPPORT is needed for CONFIG\_PROVE\_SPIN\_LOCKING and CONFIG\_PROVE\_RW\_LOCKING to be offered by the generic lock debugging code. Otherwise only CONFIG\_PROVE\_MUTEX\_LOCKING and CONFIG\_PROVE\_RWSEM\_LOCKING will be offered on an architecture - these are locking APIs that are not used in IRQ context. (the one exception for rwsems is worked around)

architecture support for this is certainly not in the "trivial" category, because lots of lowlevel assembly code deal with irq-flags state changes. But an architecture can be irq-flags-tracing enabled in a rather straightforward and risk-free manner.

Architectures that want to support this need to do a couple of code-organizational changes first:

- move their irq-flags manipulation code from their asm/system.h header to asm/irqflags.h
- rename local\_irq\_disable()/etc to raw\_local\_irq\_disable()/etc. so that the linux/irqflags.h code can inject callbacks and can construct the real local\_irq\_disable()/etc APIs.
- add and enable TRACE\_IRQFLAGS\_SUPPORT in their arch level Kconfig file

and then a couple of functional changes are needed as well to implement irq-flags-tracing support:

- in lowlevel entry code add (build-conditional) calls to the trace\_hardirqs\_off()/trace\_hardirqs\_on() functions. The lock validator closely guards whether the 'real' irq-flags matches the 'virtual' irq-flags state, and complains loudly (and turns itself off) if the two do not match. Usually most of the time for arch support for irq-flags-tracing is spent in this state: look at the lockdep complaint, try to figure out the assembly code we did not cover yet, fix and repeat. Once the system has booted up and works without a lockdep complaint in the irq-flags-tracing functions arch support is complete.
- if the architecture has non-maskable interrupts then those need to be excluded from the irq-tracing [and lock validation] mechanism via lockdep\_off()/lockdep\_on().

in general there is no risk from having an incomplete irq-flags-tracing implementation in an architecture: lockdep will detect that and will turn itself off. I.e. the lock validator will still be reliable. There should be no crashes due to irq-tracing bugs. (except if the assembly changes break other code by modifying conditions or registers that shouldnt be)

irqflags-tracing.txt