* Interruptions (asynchrone généré par HW) : interruption HW.
* Exceptions et syscalls (synchrone généré par proc et soft) : interruptions soft.

1 interrupt handler se charge de traiter une IRQ particulière et fait partie intégrante du driver : donc un interrupt handler par device.

Le contexte d’interruption est un contexte atomique : bloquant.

Plusieurs handlers peuvent être associé à une même IRQ : ils sont shared et lors d’une IRQ sur cette ligne, chaque device dit si oui ou non son device a provoqué l’IRQ : tous les interrupt handlers sont donc sondés à chaque IRQ sur cette ligne.

Interrupt handling : Top half et Bottom half :

* Les tops acquittent IRQ et refilent le taf aux bottom (exécution ultra rapide) : il s’agit des interrupt handlers (Par exemple les tops vont copier vers un buffer mémoire et laisser le traitement aux bottoms). IRQ (partiels ou totales) désactivées ici.
* Les bottoms eux procède au traitement de l’IRQ avec haut niveau de priorité mais IRQ activés

SoftIRQ, tasklets, workqueues dans cette catégorie.

Register an interrupt handler : top half

/\* request\_irq: allocate a given interrupt line \*/

int request\_irq(unsigned int irq,

irq\_handler\_t handler,

unsigned long flags,

const char \*name,

void \*dev)

Enlever l’inscription d’un interrupt handler :

void free\_irq(unsigned int irq, void \*dev)

Ecriture d’un handler :

static irqreturn\_t intr\_handler(int irq, void \*dev)

Exécution d’un IRQ :

unsigned int do\_IRQ(struct pt\_regs regs)

Il va acquitter le HW et désactiver les IRQ sur cette ligne puis appeler le bon handler si aucun n’est activé sur cette ligne puis appel handle\_IRQ\_event()

Bottom half :

1 softirqd par CPU, qui s’occupe des softirqs et tasklets.

* Softirq : contexte d’interruption

Alloués lors de la compilation, ils sont plus compétents que les tasklets lors des times criticicals les plus importants.

Un softIRQ ne peut pas préempter un autre softirq, seul un interrupt handler le peut.

Ils sont utilisés par les networking devices et les blocks devices.

* Tasklets : contexte d’interruption

Alloués dynamiquement et basé sur softirq.

* Work queues : contexte processus

Différent des deux autres.