Thread

Interrupt

hardware

pushStack(){
push;
lr = 0xFFFFFFFD;
}
exceptionTaken(){
pc = . hal_default_interrupt_vsr
CONTROL.SPSEL=0
}

hal_default_interrupt_vsr

push {lr};
add sp, #4;
...

bl hal_deliver_interrupt

hal_deliver_interrupt

pendsvc = false;
isr_result = isr(vector, hal_interrupt_data[vector]);
cyg_interrupt_post_dsr(hal_interrupt_objects[vector])
sched_lock == 0?
N
Y
pendsvc = true;
icsr |=CYGARC_REG_NVIC_ICSR_PENDSVSET

return from hal_deliver_interrupt
pc = &(add sp, #4)

add sp, #4
pop {pc}

exceptionReturn{
CONTROL.SPSEL = 1;
popStack();
}

Hardware

pushStack(){
push();
lr = 0xFFFFFFF1
}
exceptionTaken(){
pc = hal_pendable_svc_vsr;
}

hal_pendable_svc_vsr

pendSVC forges exception frame on Process Stack

bx lr;

exceptionReturn(){
popStack()
}

return from hal_deliver_interrupt
pc = &(add sp, #4)

add sp, #4
pop {pc}

exceptionReturn_1(){
CONTROL.SPSEL = 1
}

exceptionReturn_2(){
popStack(){
pc = .hal_interrupt_end;
lr = .hal_interrupt_end_done;
}
}

hal_interrupt_end

Another thread

sched_lock++;
interrupt_end(0,0,0)
unlock()
unlock_inner start
Switch context
Switch context
unlock_inner finish

return from hal_interrupt_end{
pc = hal_interrupt_end_done;
}

hal_interrupt_end_done

swi 0;

hardware

pushStack(){
lr = 0xFFFFFFFD;
}
exceptionTaken(){
pc = hal_interrupt_end_VSR;
CONTROL.SPSEL = 0
}

hal_interrupt_end_VSR

hal_interrupt_end_VSR discards frame of "swi 0"

bx lr;

exceptionReturn(){
CONTROL.SPSEL = 1;
popStack();
}

0
psp Process Stack msp Main Stack

1
exception frame of thread 8 words
psp msp

2
exception frame of thread 8 words
lr=0xFFFFFFFD
psp msp

3
exception frame of thread 8 words
lr = 0xFFFFFFFD
local stack of hal_deliver_interrupt
psp msp

4
exception frame of thread 8 words
lr = 0xFFFFFFFD
local stack of hal_deliver_interrupt
exception frame of hal_deliver_interrupt 8 words
psp msp

5
exception frame of thread 8 words
fake frame to call interrupt_end 8 words
lr = 0xFFFFFFFD
local stack of hal_deliver_interrupt
exception frame of hal_deliver_interrupt 8 words
psp msp

6
exception frame of thread 8 words
fake frame to call interrupt_end 8 words
lr = 0xFFFFFFFD
local stack of hal_deliver_interrupt
psp msp

7
exception frame of thread 8 words
fake frame to call interrupt_end 8 words
lr = 0xFFFFFFFD
psp msp

8
exception frame of thread 8 words
fake frame to call interrupt_end 8 words
psp msp

9
exception frame of thread 8 words
fake frame to call interrupt_end 8 words
psp msp

10
exception frame of thread 8 words
psp msp

11
exception frame of thread 8 words
local stack of hal_interrupt_end
psp msp

12
exception frame of thread 8 words
psp msp

13
exception frame of thread 8 words
psp msp

14
exception frame of thread 8 words
frame of hal_interrupt_end_done 8 words
psp msp

15
exception frame of thread 8 words
psp msp

16
psp msp