1.

sys\_thrdstop(void):

First, Initialize interval, handler, ticks in p.  
Check if thrdstop\_context\_id from input < 0, find an empty thrd\_context\_data space for it.

Return the new thrdstop\_context\_id

sys\_cancelthrdstop(void):

Reset the interval in p to -1 to cancel thrdstop

Check if thrdstop\_context\_id from input >=0,

store data from trapframe to thrd\_context\_data[thrdstop\_context\_id]

Return ticks in p

sys\_thrdresume(void):

Check if is\_exit from input == 0, store data from thrd\_context\_data[thrdstop\_context\_id] to trapframe.

Else reset inverval in p = -1 to cancel thrdstop and turn off p->thrdstop\_context\_used[thrdstop\_context\_id]

2.

先存下當前的環境參數，包含s\_regs[12], ra, sp, t\_regs[7], a\_regs[8], gp, tp, epc, 再去swtch到其它地方，這樣以利之後再swtch回來這個被儲存的環境。 可以先只存callee的環境參數，不用存caller的環境參數，等要離開caller到下一個環境時再存就好。因為之後被handler swtch回到該環境時，只需要用到callee的環境參數，所以多存一組caller參數是不必要的。

3.

因為原本在callee的環境裡，環境要被caller的環境取代了，為了離開以後還能回到callee，必須先儲存callee的環境參數

，而caller的環境參數可以等到要離開那個caller的環境再存，這樣可以節省資源。