OSI Handout Week03.a: Context Switch

1. Context switch in user-space:

a) **void ctx_start(void** old_sp, void* new_sp);**

This will be used when starting a new thread.
It will save registers on the old stack,
store current stack pointer to "old_sp",
switch stack to the "new_sp",
and
finally call ctx_entry().

b) **void ctx_switch(void** old_sp, void* new_sp);**

This will be used for context switch.
It will save registers on the old stack,
store current stack pointer to "old_sp",
switch stack to the "new_sp",
and
restore registers from the new stack,
finally return (to ra).

2. user/apps/thread.s

```
1       .macro SAVE_REGISTERS
2           sw s0,4(sp)       /* Save callee-saved registers */
3           sw s1,8(sp)
4           sw s2,12(sp)
5           sw s3,16(sp)
6           sw s4,20(sp)
7           sw s5,24(sp)
8           sw s6,28(sp)
9           sw s7,32(sp)
10          sw s8,36(sp)
11          sw s9,40(sp)
12          sw s10,44(sp)
13          sw s11,48(sp)
14          sw ra,52(sp)      /* Save return address */
15      .endm
16
17      .macro RESTORE_REGISTERS
18          lw s0,4(sp)       /* Restore callee-saved registers */
19          lw s1,8(sp)
20          lw s2,12(sp)
21          lw s3,16(sp)
22          lw s4,20(sp)
23          lw s5,24(sp)
24          lw s6,28(sp)
25          lw s7,32(sp)
26          lw s8,36(sp)
27          lw s9,40(sp)
28          lw s10,44(sp)
29          lw s11,48(sp)
30          lw ra,52(sp)      /* Restore return address */
31      .endm
32
33      /* void ctx_start(void** old_sp, void* new_sp); */
34      /*                        ^             ^       */
35      /*                        |             |       */
36      /*                        a0            a1      */
37      ctx_start:
38          addi sp,sp,-64
39          SAVE_REGISTERS
40          sw sp,0(a0)       /* Save the current stack pointer */
41          mv sp,a1          /* Switch the stack */
42          call ctx_entry    /* Call ctx_entry() */
43
44      /* void ctx_switch(void** old_sp, void* new_sp); */
45      /*                         ^             ^       */
46      /*                         |             |       */
47      /*                         a0            a1      */
48      ctx_switch:
49          addi sp,sp,-64
50          SAVE_REGISTERS
51          sw sp,0(a0)       /* Save the current stack pointer */
52          mv sp,a1          /* Switch the stack */
53          RESTORE_REGISTERS
54          addi sp,sp,64
55          ret
```

3. An example use of ctx_start+ctx_entry

```
void thread_create(void (*f)(void *), void *arg, unsigned int stack_size) {

  tcb = create_thread_control_block();
  old_tcb = current_running_thread_control_block();

  … // do something necessary

  void **old_sp = …   // old stack pointer's address in old_tcb
  void *new_sp =  …   // new stack pointer in tcb

  ctx_start(old_sp, new_sp);
}


void ctx_entry(void){

    … // do something useful

    (*f)(arg);  // run function "f" received by "thread_create"

    … // wrap up
}
```