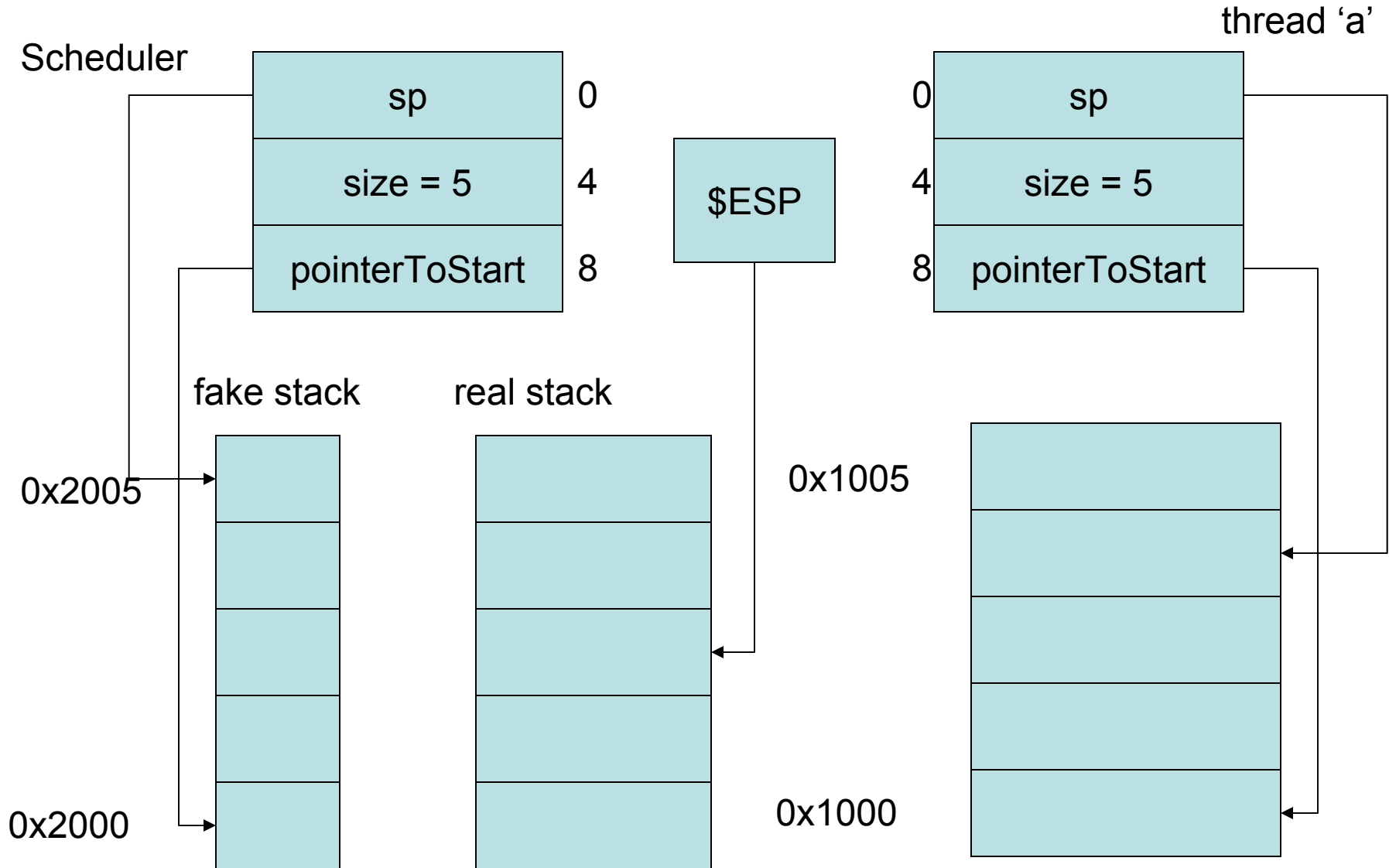


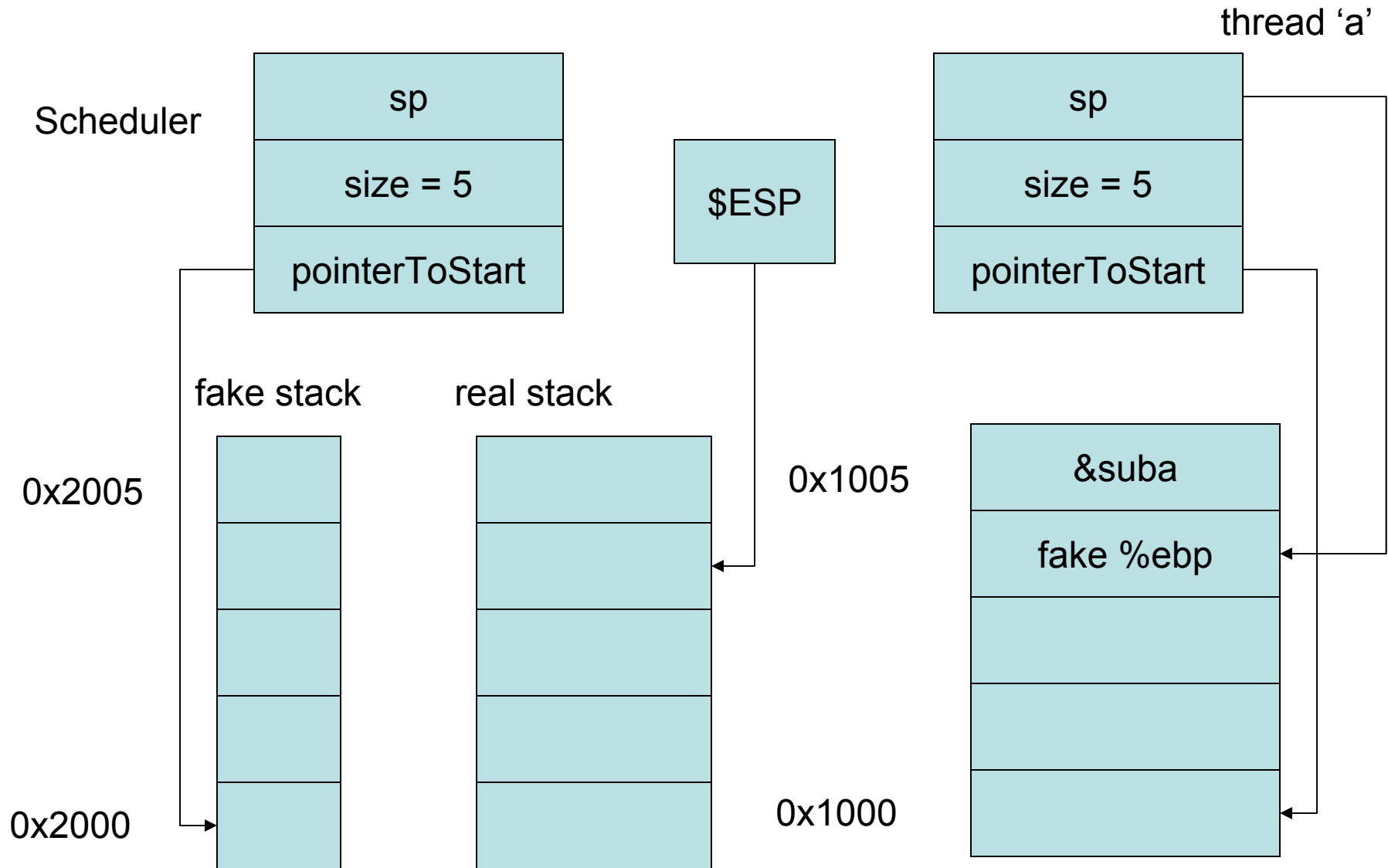
# Doing the context switch

- \$ESP is the real stack pointer, the one in the machine. There's only one. It moves from the "real stack" to your "thread stack" and back again
- The offset #'s on the MyThread data structures are the offsets of those fields
- The Scheduler creates a "fake stack" that isn't really used – when doing a switch from Scheduler to thread 'a', the value stored in Scheduler.sp will be the original "real stack" pointer. It should "look different" than the sp's in your thread stacks
- A procedure call ALWAYS's pushes a program counter address. It MAY push extra stuff. In my example, I assume that register %ebp is pushed by contextswitch. It doesn't have to be. You may also push more stuff – what matters is that your stack, when initialized, have *all* the stuff on it that contextswitch is going to pop/remove

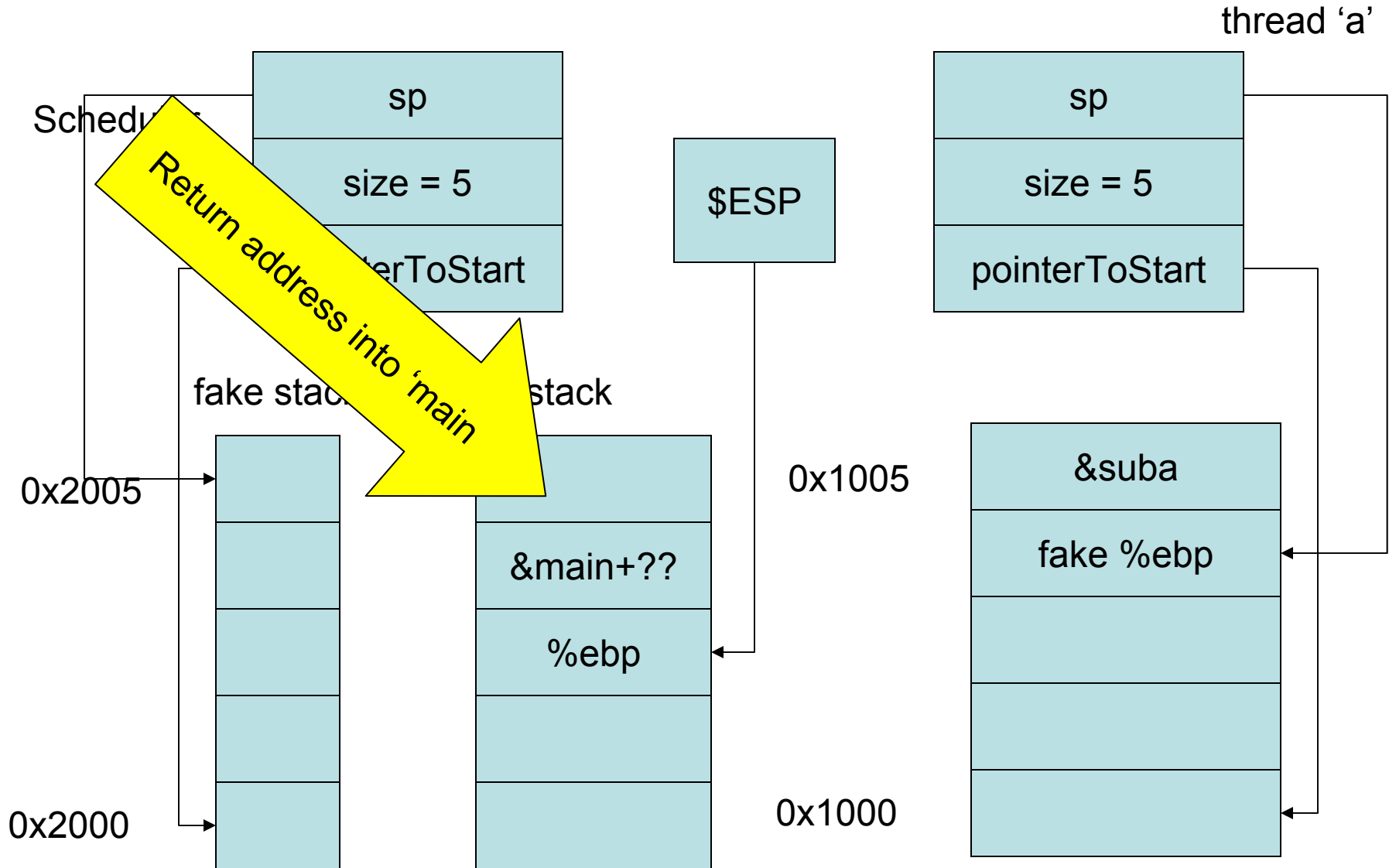
# Before setStart



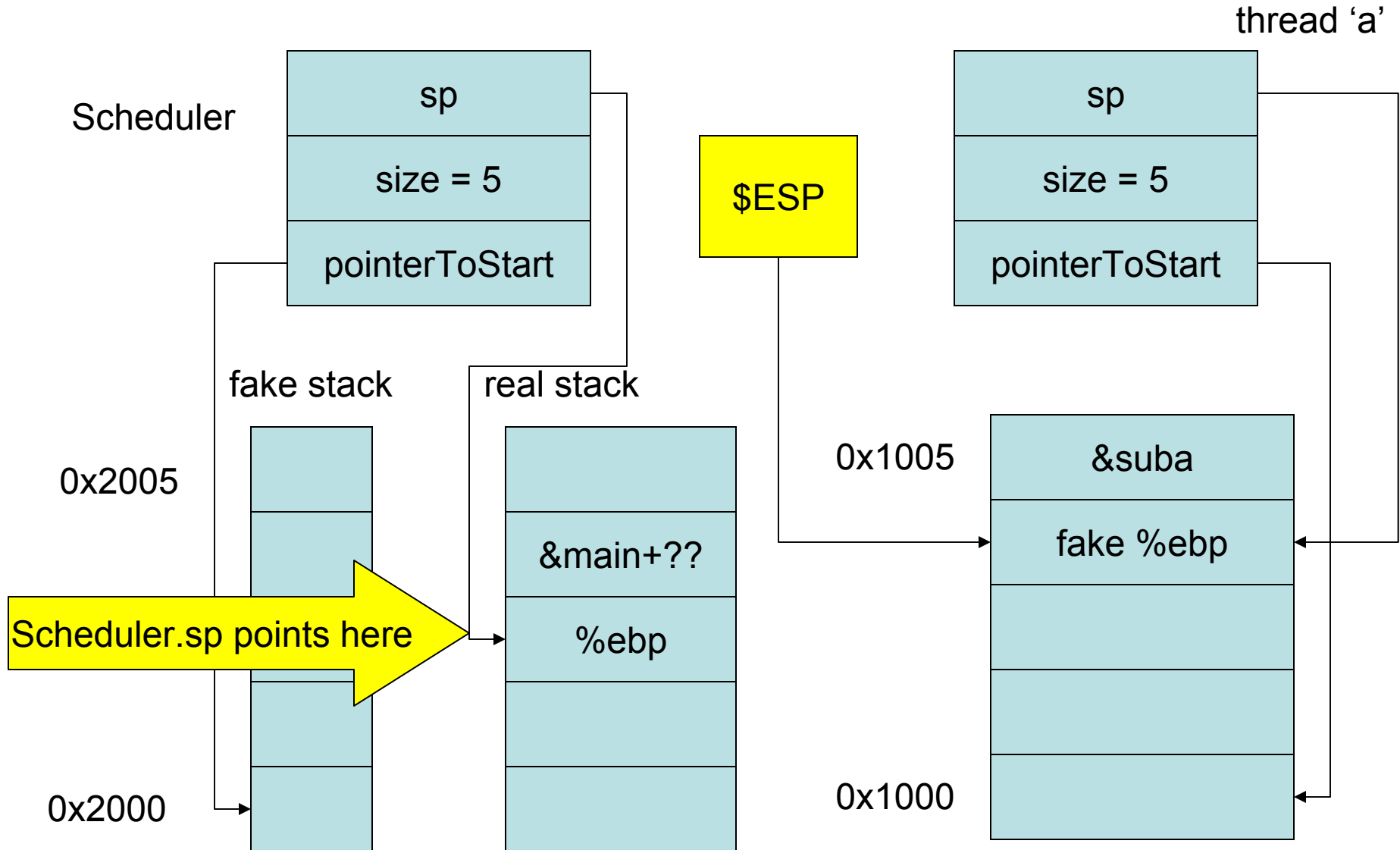
# After setStart, before context switch



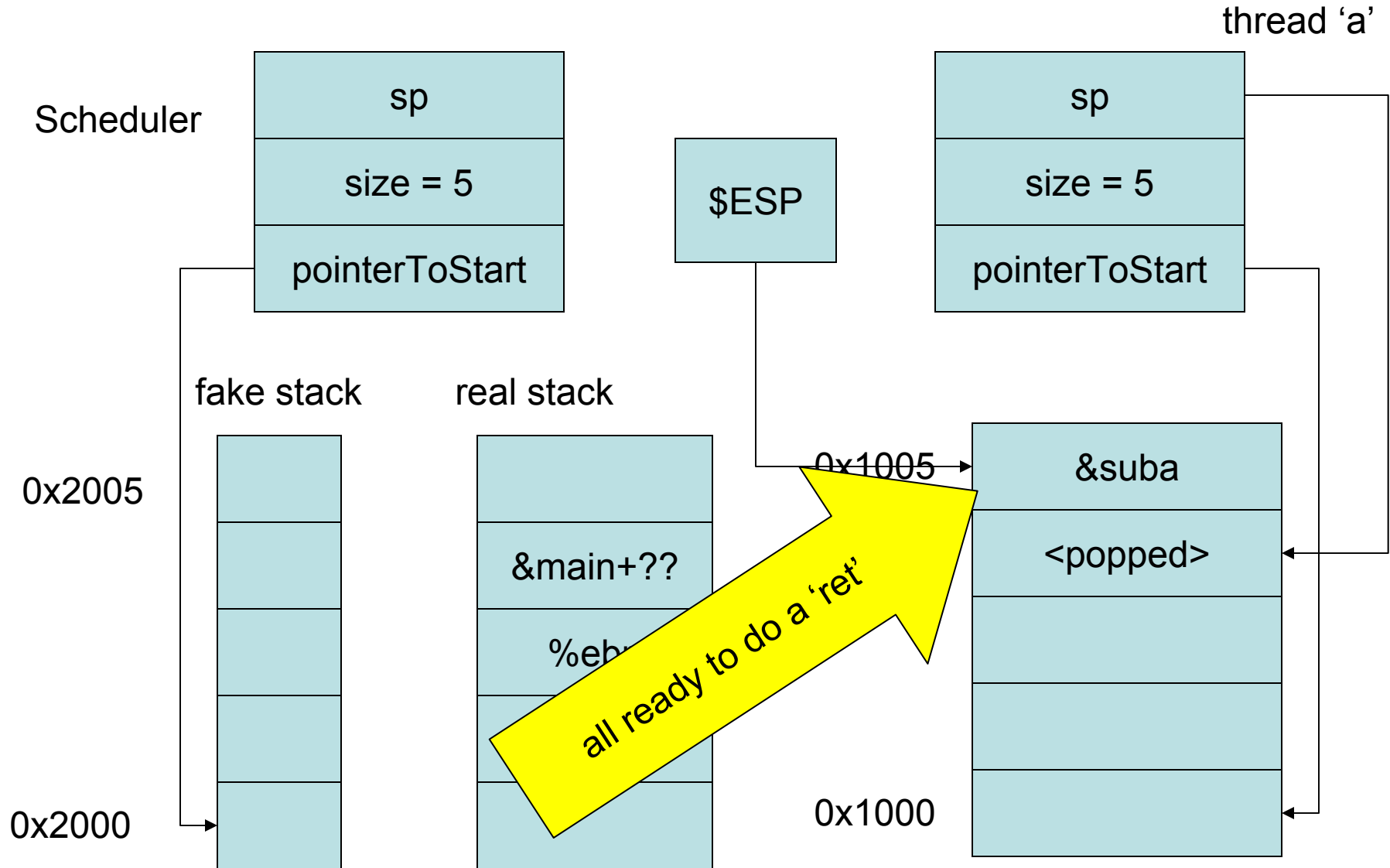
# Middle of context switch



# Middle+1 of context switch



# Exiting contextswitch



# More notes:

- The SP (e.g. Scheduler.sp, a.sp, b.sp) should point to the last item on the stack **HOLDING VALID DATA**.
  - It does not point one past the last valid data!
- Note:
  - `*--sp = foo` means “decrement sp, assign foo to \*sp”
  - `*sp-- = foo` means “assign foo to \*sp, decrement sp”
  - NOTE: my code uses `*--sp`; Wang Tin Ling may have seen this as `*sp--`, causing some confusion.
- To dump out the assembly from a ‘.o’ file (e.g. First.o)
  - `objdump -disassemble First.o`