

NOTE:

All functionalities are tested and working well.

In part 3, 'ubuffer' that was passed to the second argument of `cbregister()` is a global variable served as user-space message buffer. Kernel will copy the content of temporary message buffer into ubuffer, which callback function uses to print the message.

You can speculate `sleepReceiver()` and `clkdispReceiver()` in `main.c` to have a closer look of my design.

New files added: `cbregister.c`, `memList.c`

Bonus question:

Before a process is context switched out, we can save its original return address state into a temporary buffer. And before it was context switched in again, we can implement a similar thing as we do in `part3` in `ctxsw()` (and similarly for `resched()` and `sleepms()`). In `part3`, we check if a callback function is registered, if so we modify the return address.

So here, we check if the return address is modified and thus differed from original return address saved in the buffer. If so, we modify the return address to be the original one.