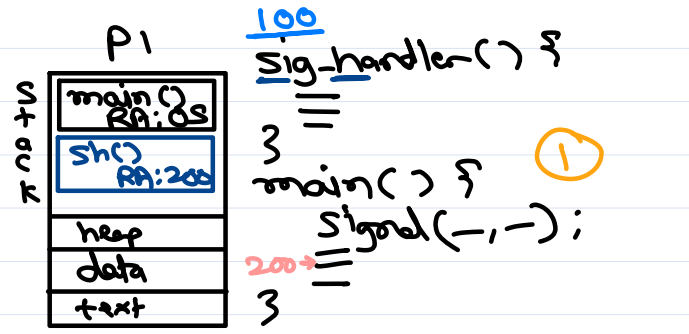
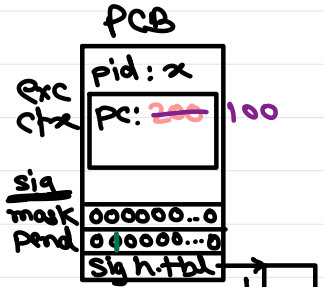
Embedded Operating Systems

Trainer: Nilesh Ghule

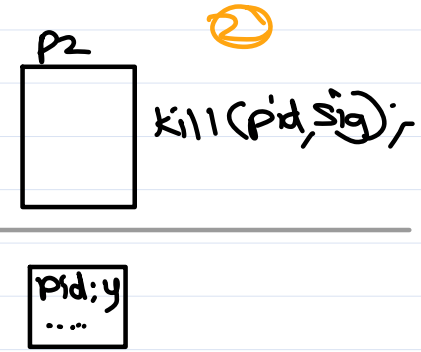




③ intr_handler:
1. Save exec ctx.
2. get ISR addr
3. call ISR
4. pid = scheduler();
5. sig_delivery(pid);
6. restore exec ctx.
   dispatcher(pid).

sys_signal(sig, hand)
1. get current process pcb.
2. write hand fn addr in its sig handler tbl slot corresponding to sig num.

sig_delivery(pid):
1. in pcb of given process check pending but non-masked signals.
2. for each such signal:
   a. create artificial stack frame on user stack for that signal's handler.
   b. the return addr of the stack frame will be the addr from exec ctx (when process should resume after handler completion).
   c. change PC in exec ctx of the process to the signal hand fn addr (obtained from sig handler table).
   d. clear sig pending bit.



sys_kill(pid, sig):
1. get pcb of the process from process list whose pid is given.
2. Set the bit corresponding to signal number in sig pending field of that pcb.

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>