First RTOS

Sample Implementation

Group Members:

Sinem Coleri

Slobodan Matic

Anshuman Sharma

Model

- Important to distinguish between user space (*applic.c*) vs. kernel space (*emachine.c*)
- Makes code more portable, modular
- Ecode provides task schedule specification
- Emachine
  - part of the kernel
  - Periodically polls enabled triggers for activation (for LegOS - once every 20 ms)

Emachine

```
while (i < n_enabled_triggers) {
   if (e_schedule[i].trigger_time <= sys_time) {
    pc = e_schedule[i].address;
     …
     while (!end) { …
             switch(e_code[pc].opcode) {
             case OPCODE_FUTURE:
                          /*enable, insert and set trigger_time */
             case OPCODE_CALL:
                          /* execute driver_code*/
             case OPCODE_SCHEDULE:
                          /* post task-specific semaphore */
             case OPCODE_RETURN:
                          /* end == 1*/
             }
     }
```

Task Activation

- All tasks are created as part of initialization
- Task activation via semaphores

```
int task_code() {
        while(1) {
                sem_wait(task_semaphores[.]);
                /*execute task-body */
        }
}
```

Interface

- Providing the interface between user and kernel space
- e_machine_init()
  - instruction_t *program
  - sem_t *sem
  - driver_code_t *driver
- Called for each time new Ecode is executed

Tasks

- Poll light sensor @ 2Hz and display on lcd
- Turn on and off light sensor
- Beep @ 1Hz

Ecode

```
instruction_t program[MAXINSTR] = {
  /* 0 */  CALL(0),                    /*light sensor read */
  /* 1 */  SCHEDULE(0),                /* writing to LCD */
  /* 2 */  SCHEDULE(1),                /* beeping */
  /* 3 */  FUTURE(500,5),              /* enabling trigger */
  /* 4 */  RETURN(),                   /* finish with Ecode */
  /* 5 */  CALL(0),
  /* 6 */  SCHEDULE(0),
  /* 7 */  FUTURE(500,0),
  /* 8 */  RETURN()
};
```