PROGRAMMING CONCEPTS AND EMBEDDED PROGRAMMING IN C, C++ and JAVA:

Lesson-9: Programming using Event or Messages Polling

1. Multitasking

Function *main* with a waiting loop

- main ( ) passes the control to an RTOS
- Each task controlled by RTOS and
- Each task will also have the codes in an infinite loop
- A waiting task is passed a signal by the RTOS to start

main ( ) calling RTOS

- # define false 0
- # define true 1
- /**********************************
  **********/
- void main (void) {
- /* Call RTOS run here */

Infinite loop in main ( )

- while (1) {rtos.run ( );
- /* Infinite while loops follows in each task. So never there is return from the RTOS. */
- }
- }
- /**********
  **********/

Task 1

- void task1 (....) {
- /* Declarations */
- .
- while (true) {
- /* Codes that repeatedly execute */
- .
- /* Codes that execute on an event */
- if (flag1) {....;}; flag1 =0;
- /* Codes that execute for message to the kernel */
- message1 ( );
- } }
- /**/

Task2 ( )

- void task2 (....) {
- /* Declarations */
- .
- while (true) {
- /* Codes that repeatedly execute */
- .
- /* Codes that execute on an event */
- if (flag2) {....;}; flag2 =0;
- /* Codes that execute for message to the kernel */
- message2 ( );
- } }
- /**/

TaskN_1 ( )

- void taskN_1 (....) {
- /* Declarations */
- .
- while (true) {
- /* Codes that repeatedly execute */
- .
- /* Codes that execute on an event */
- if (flagN_1) {....;}; flagN_1 =0;
- /* Codes that execute for message to the kernel */
- messageN_1 ( );
- } }
- /**/

TaskN

- void taskN (....) {
- /* Declarations */
- .
- while (true) {
- /* Codes that repeatedly execute */
- .
- /* Codes that execute on an event */
- if (flagN) {....;}; flagN =0;
- /* Codes that execute for message to the kernel */
- messageN ( );
- } }
- /**/

2. Polling for events and messages

Polling for events and messages

- A Programming method is to facilitate execution of one of the multiple possible function calls and the function executes after polling
- Polling example is polling for a screen state (or Window menu) *j* and for a message *m* from an ISR as per the user choice

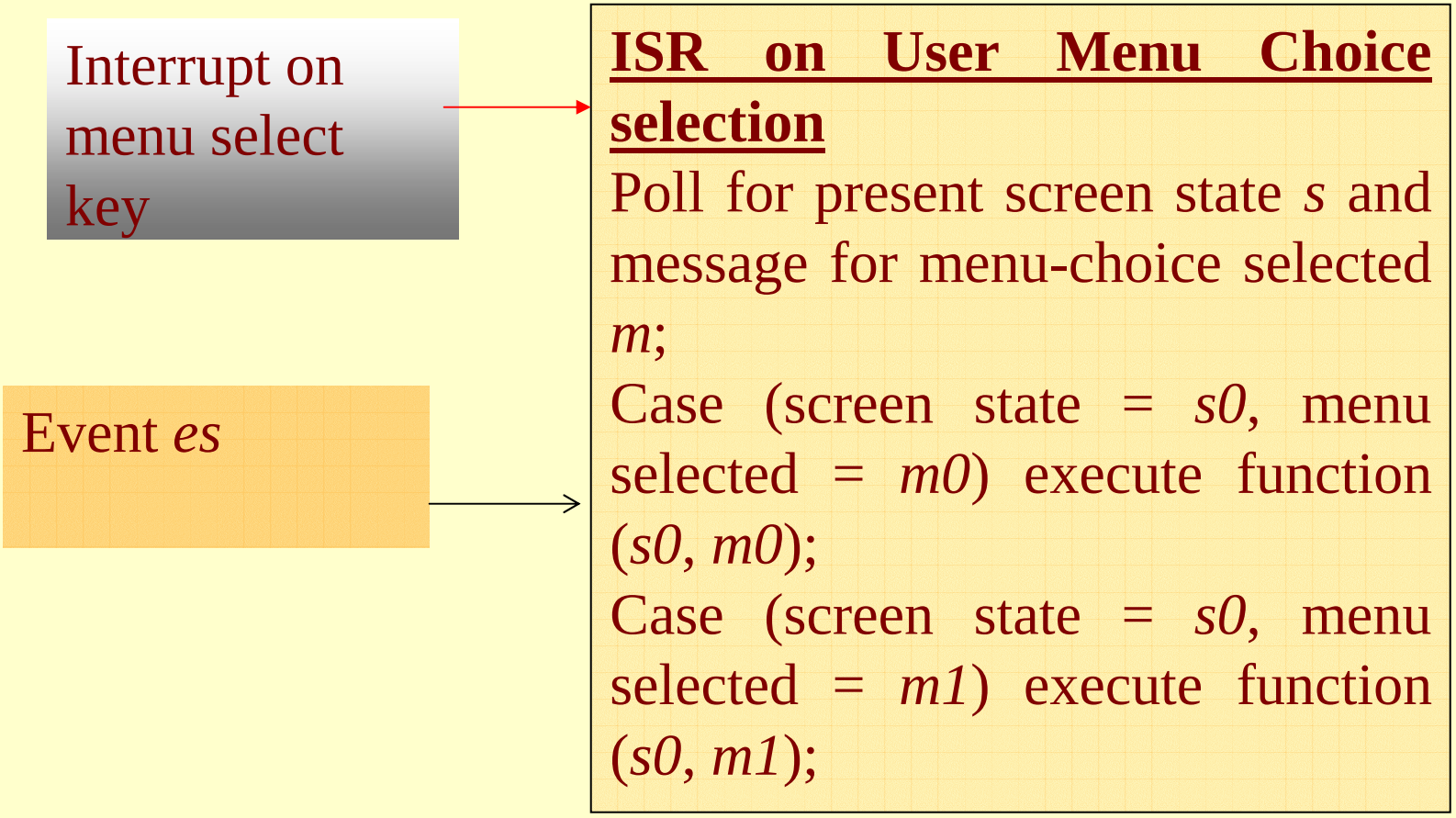
Mobile phone

- Assume that screen state j is between 0 and K, among 0, 1, 2, .. or K – 1 possible states.(set of menus).
- An interrupt is triggered from a touch screen GUI and an ISR posts an event-message *m* = 0, 1, 2, …, or N – 1 as per the selected the menu choice 0, 1, 2, …, N – 1 when there are N menu- choices for a mobile phone user to select from a screen in state j.

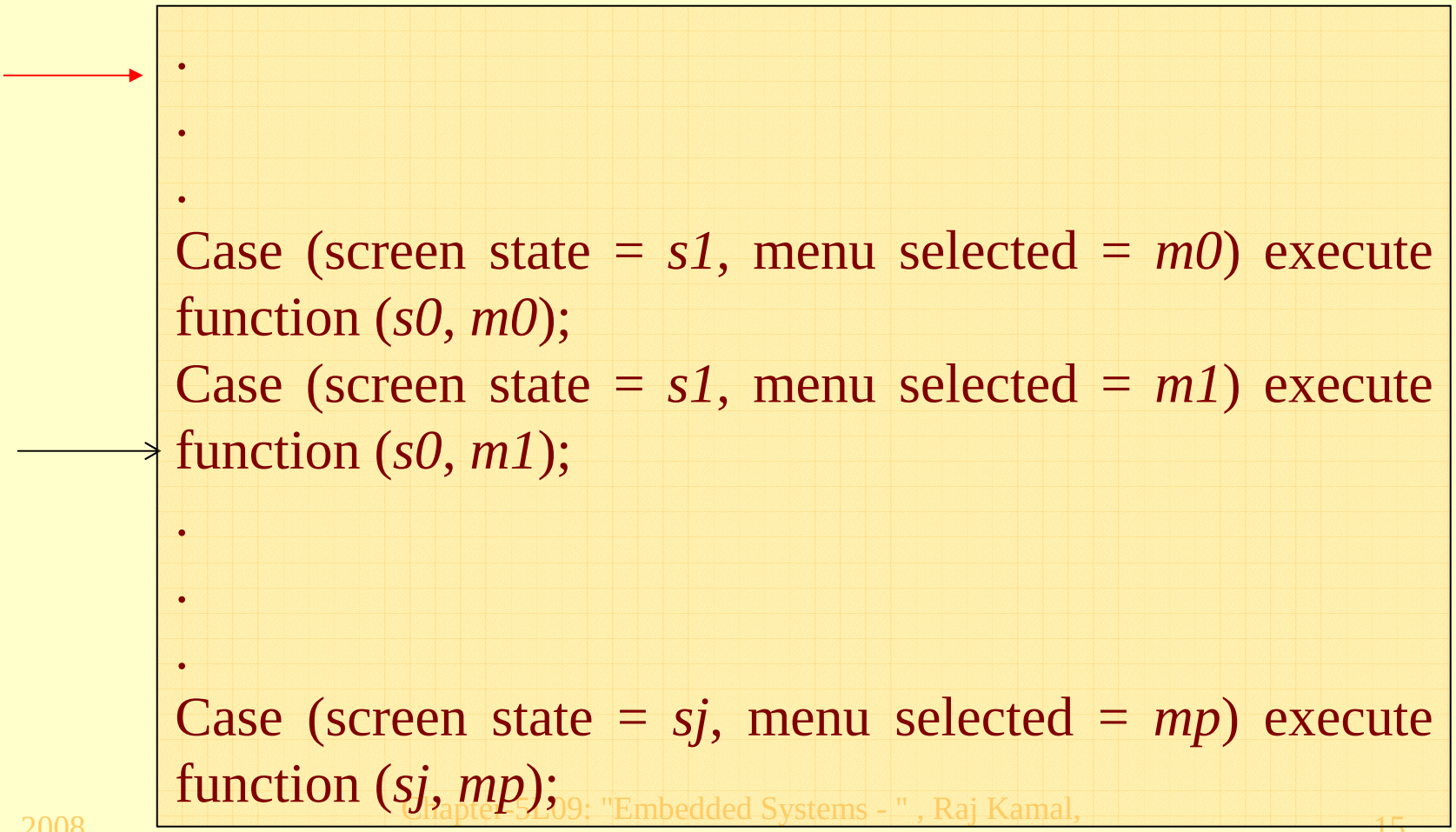
Polling for a menu selection from screen state

- void poll_menuK {/* Code for polling for choice from menu m for screen state K*/
- }
- }
- /**********************************/

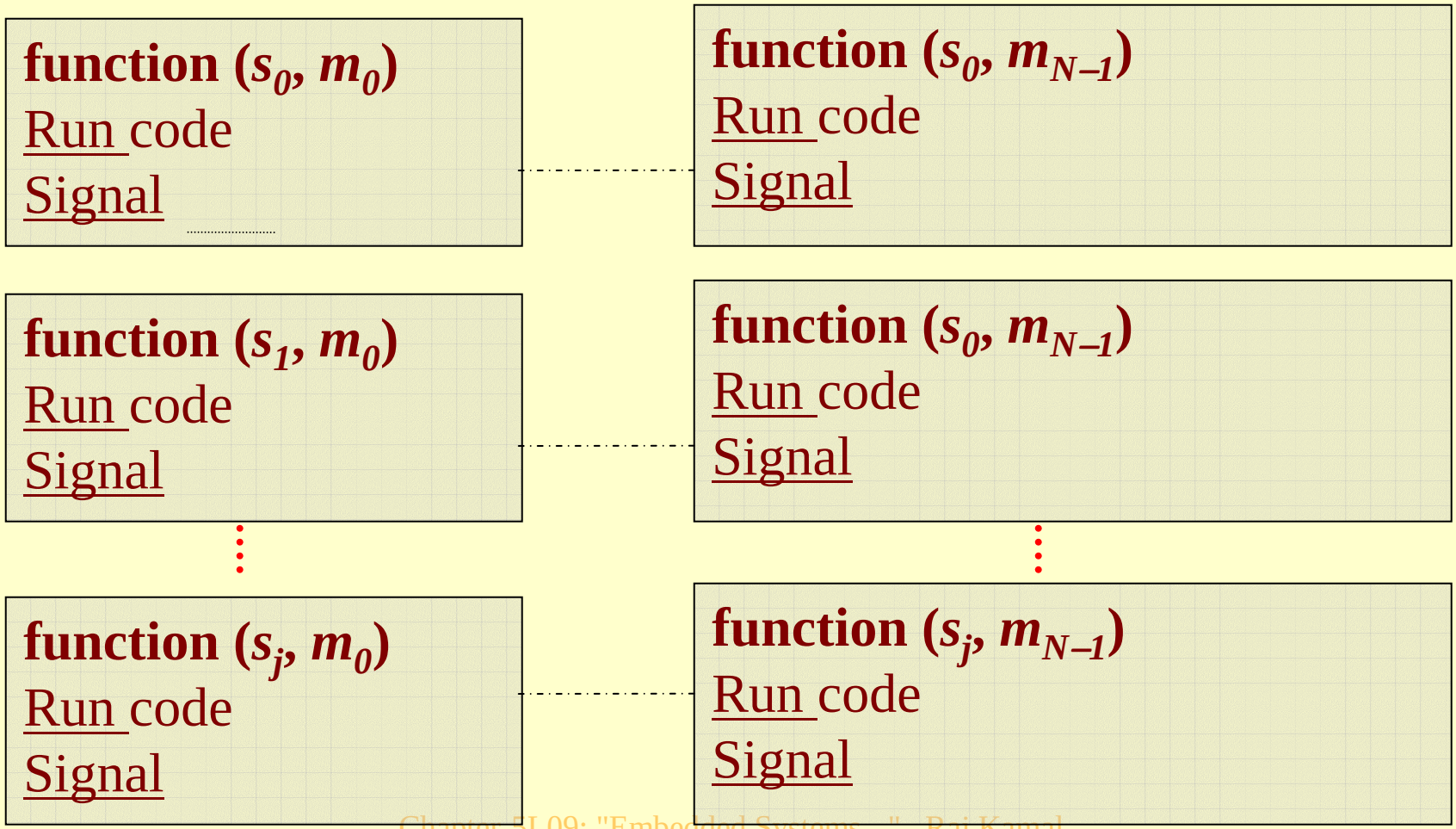
Polling for an even like mice click or menu select



Polling for an even like mice click or menu select



One of Functions (s, m) called



Summary

We learnt

- A program function can be initiated by polling for a signal or message sent by the operating system
- A signal or message sent by the operating system can be as per screen state or window menu and user choice

End of Lesson 9 of Chapter 5