

January 2010 - c

SORTES: Software for embedded and real-time systems

- 1) The elements of a linked list include a pointer to the next element in the list and a pointer to a data structure.

You are asked to implement in C two functions for a *filtered* linked list. Filtered means here that an element should be in the list if and only if it satisfies a requirement verified by some function associated to the list when the list is initialised.

You must provide the C code to define:

- a structure called `f_list` including a pointer to the first element of the list and a pointer to the filtering function (taking `my_data` as parameter and returning TRUE if the requirements are satisfied and FALSE if not);
- a structure called `f_elem` including a pointer to the next element of the list and a pointer to the data structure of type `my_data`.

You must also provide the declaration and the definition of the following functions:

- `f_list_init`: receives a filtering function as parameter and a pointer to the first element of a linked list and creates an filtered list. It returns the number of elements in the list.
- `f_list_filter`: receives a `f_list` as parameter. The function removes from the list the elements that are rejected by the filtering function. The function returns the number of elements remaining in the list

The implementation must work with any type of data and any filter.

- 2) Represent in ASG the behaviour of the following system, including a controlled device, two controllers and an arbitrator. Every second, the system goes through the following cycle: the controlled device sends an identical status message to the two controllers. Each controller takes a decision and sends a decision message to the arbitrator. If the two decisions are identical, the arbitrator applies the decision to the controlled device, else it tells the controllers to decide again. If at least two of the 4 decisions produced in the cycle match (and if only 2, the other 2 do not match), it is applied to the controlled device and an alarm a is raised if a same controller failed two times. If the 4 decisions are different or if the 2 controllers produced twice the same different decision, the system stop and alarm b is raised.
- 3) If you implement an ASG diagram with techniques similar to the Microchip TCP-IP stack, how would you implement
 - a rendez-vous involving 3 components
 - a resource

Explain of course.

- 4) Neither MicroC/OS-II nor the Microchip TCPIP stack support variable priorities. Explain how the two scheduling methods differ and what to add in programs in each case to force the behaviour of the other.

duration: 3 hours, with open books and notes.