

```
// [...]  
  
// EXTENSION 2016-09-20  
  
/**  
 * In this new implementation we will be taking the one provided in  
 * 2016-09-20_22 one step further by introducing broadcasting roles  
 * and allowing for multiple broadcast messages: the system will store  
 * all broadcasting messages (which can be sent only by proceseses  
 * registered to the global broadcast as broadcasters) until a maximum  
 * value defined in costanti.h.  
 */  
  
/**  
 * Broadcasting role: each process can register to the global system broadcast  
 * either as a broadcaster or a listener. When a process is created in the  
 * system module its role is set to B_NONE.  
 */  
extern "C" enum broadcast_role  
{  
    B_BROADCASTER = 1,  
    B_LISTENER  
};  
  
/**  
 * Registers the current process as a listener of the global broadcast with the  
 * given role. The calling process must be aborted if the specified role is not  
 * one between broadcaster or listener as well as if the process is already  
 * registered to the global broadcast or there is already a broadcaster process  
 * registered.  
 *  
 * @param role the broadcast role to be used for the process registrations.  
 */  
extern "C" void reg(enum broadcast_role role);  
  
/**  
 * Returns to the calling process the next broadcast message. If the process has  
 * already retrieved all available broadcast messages it will be placed in the  
 * listeners wait queue. All processes in this queue will be rescheduled when a  
 * new broadcast message is sent.  
 */  
extern "C" natl listen();  
  
/**  
 * Sends the given broadcast message using the system global broadcast  
 * descriptor. The calling process must be aborted if it is not the currentl  
 * registered broadcaster.  
 *  
 * @param msg the broadcast message to be sent.  
 */  
extern "C" void broadcast(natl msg);  
  
// EXTENSION 2016-09-20
```