

```
// [...]  
  
// EXTENSION 2019-07-24  
  
/**  
 *  
 */  
typedef natq vaddr;  
  
/**  
 * Primitives declarations for the User Module: we will be providing to the  
 * processes the ability to pause a certain process when it reaches a certain  
 * instruction.  
 */  
  
/**  
 * A master process can use to primitive to set a breakpoint (assembly  
 * instruction int3, opcode 0xCC) at the given instruction address for the slave  
 * process having the given id. If a process executes the int3 instruction  
 * without being a slave it must be aborted.  
 *  
 * The specified process must not be a slave or a master process itself.  
 *  
 * In case of error the calling process must be aborted.  
 *  
 * @param id id of the slave process;  
 * @param rip process instruction address: keep in mind that it must be in the  
 * interval defined by [ini_utn_c, fin_utn_c);  
 *  
 * @return true if the breakpoint is successfully installed, false otherwise.  
 */  
extern "C" bool bpattach(natl id, vaddr rip);  
  
/**  
 * The master process can suspend its execution and wait until the breakpoint in  
 * the slave process is reached. The calling process must be a master process.  
 *  
 * In case of error the calling process must be aborted.  
 */  
extern "C" void bpwait();  
  
/**  
 * Undoes the operation done by bpattach() and reschedules the slave process.  
 * The calling process must be a master process.  
 *  
 * In case of error the calling process must be aborted.  
 */  
extern "C" void bpdetach();  
  
// EXTENSION 2019-07-24  
  
// [...]
```