

```
// [...]  
  
// EXTENSION 2019-06-12  
  
/**  
 * Virtual address definition for the User Module.  
 */  
typedef natq vaddr;  
  
/**  
 * We want to provide to the processes a mechanism to allow them to know if the  
 * execution of one of the other processes reaches a certain instruction. Only  
 * the first process reaching the breakpoint instruction must be intercepted.  
 * All other processes should work seamlessly.  
 */  
  
/**  
 * This primitive allows a user process to install a breakpoint (assembly  
 * instruction int3, opcode 0xCC) at the address pointed by rip. The calling  
 * process (P1) must be paused. When another process (P2) reaches the breakpoint  
 * address P1 must be rescheduled. The P2 process is not paused, however it  
 * might have to be rescheduled in order to guarantee processes priority.  
 *  
 * All other process executing the int3 instruction must be aborted.  
 *  
 * The rip address must belong to the process user/shared memory area. Otherwise  
 * the calling process must be aborted.  
 *  
 * @param rip breakpoint instruction address.  
 *  
 * @return the id of the intercepted process at the given address or 0xFFFFFFFF  
 *         if another process has already requested a breakpoint (at any given  
 *         address).  
 */  
extern "C" natl breakpoint(vaddr rip);  
  
// EXTENSION 2019-06-12  
  
// [...]
```