

```
// [...]  
  
// EXTENSION 2019-07-03  
  
/**  
 * Virtual address definition for the User module.  
 */  
typedef natq vaddr;  
  
/**  
 * Primitives declarations for the User Module. We want to provide to the User  
 * processes the ability to pause the execution of all processes which execute  
 * a certain instruction.  
 */  
  
/**  
 * Allows to install a breakpoint at the instruction having the provided virtual  
 * address (keep in mind that the breakpoint exception has assembly instruction  
 * int3 and opcode 0xCC). Once this method is called on a given address, all  
 * processes execution will be paused and queued.  
 *  
 * For simplicity, at any given time, there is one and only one breakpoint  
 * installed. If a process spontaneously calls the int3 instruction without  
 * being added using this primitive, the calling process must be aborted.  
 *  
 * @param rip the virtual address of the instruction to be replaced with the  
 * breakpoint;  
 */  
extern "C" bool bpadd(vaddr rip);  
  
/**  
 * Can be used by a process to wait until a process reaches the breakpoint  
 * instruction address. This primitive waits for one process to reach the  
 * breakpoint instruction address and returns its id; it can be called multiple  
 * times in order to retrieve the id of all the processes with the breakpoint  
 * installed; it is an error if there not breakpoints installed in the system  
 * global breakpoint descriptor.  
 */  
extern "C" natl bpwait();  
  
/**  
 * Removes the breakpoint and reschedules all paused process. The processes  
 * rescheduled must execute without side effects.  
 */  
extern "C" void bpremove();  
  
// EXTENSION 2019-07-03  
  
// [...]
```