```
// [..]

// EXTENSION 2017-02-07

/**
 * We want to provide to the kernel the ability to allow user processes to make
 * their virtual pages permanent. To this end, we will provide the resident()
 * primitive which will make the addressed virtual pages permanent until the
 * nonresident() primitive is used to undo the operation.
 */

/**
 * User Primitives declarations.
 */

/**
 *
 */
extern "C" natq countres();

/**
 * Makes permanent all virtual pages starting from address base (inclusive) till
 * address base+size (exclusive). If the primitive execution succedes not more
 * page fault can happen for the addressed virtual pages (since there are
 * permanently stored in the physical memory) until the nonresident() is called.
 * This means that the primitive will also take care of loading into main memory
 * the missing pages. This might also be the reason of a failure if there is not
 * enough memory.
 *
 * @param  base  virtual pages starting address;
 * @param  size  virtual pages length;
 *
 * @return  the id to be given to the nonresident() primitive to undo the
 *          operation or 0xffffffff in case of failure.
 */
extern "C" natl resident(addr base, natq size);

/**
 * Reverts the operation performed using the resident() primitive.
 *
 * @param  id  the id of the operation to be undone.
 */
extern "C" void nonresident(natl id);

// EXTENSION 2017-02-07

// [...]
```