# Hello

### kern/main/hello.c

#include <types.h>  
#include <lib.h>  
void hello(void); // prototipo  
void hello(void) {  
 kprintf("Hello from me!\n");  
}

### kern/main/main.c

#include "opt-hello.h"  
#if OPT\_HELLO   
void hello(void); //prototipo va sempre dichiarato  
#endif

… dentro kmain():

#if OPT\_HELLO   
 hello(); // chiamata a hello  
#endif

### kern/conf/conf.kern

defoption hello  
optfile hello main/hello.c

### kern/conf/HELLO (creato come copia di DUMBVM)

options hello

# Locks

### kern/include/synch.h

Proprietà aggiunte a struct lock:

* int locked: valori 0/1 per indicare stato del lock
* struct wchan \*lock\_wchan: canale per dormire
* struct spinlock lock\_lock: spinlock per proteggere variabili dentro il lock
* volatile struct thread \*thread\_with\_lock: serve per memorizzare qual thread ha il lock (serve nella funzione lock\_do\_i\_hold()). È volatile perché così il compilatore ottimizzando non fa casino??

### kern/thread/synch.c

Modificate le primitive del lock (vedere codice sorgente)