Conceptes Avançats de Sistemes Operatius

Facultat d'Informàtica de Barcelona
Dept. d'Arquitectura de Computadors

Curs 2013/14 Q1

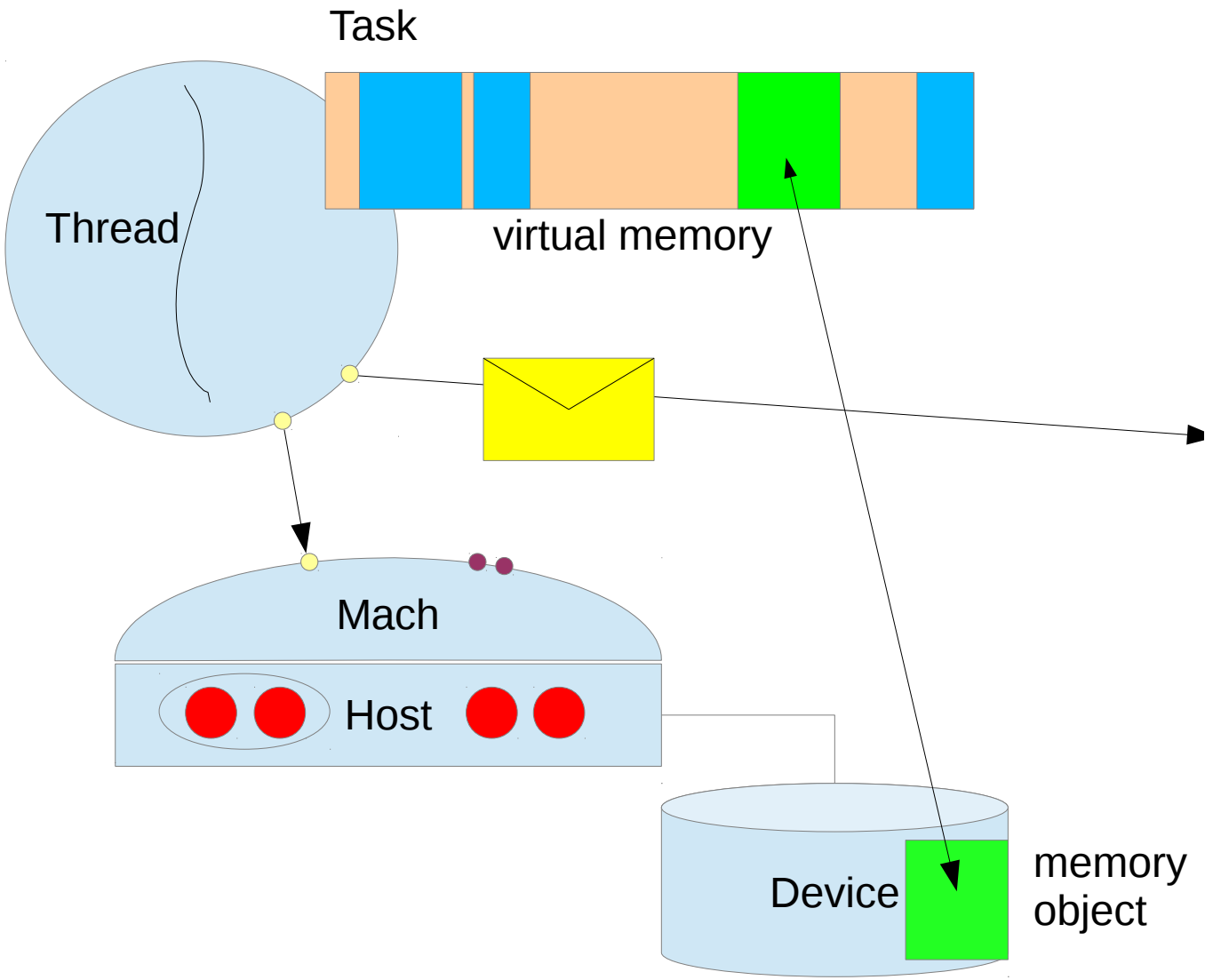
Mach / eines / suport hardware

Índex

- Interfície de Mach
- Eines de desenvolupament
  - gcc/binutils
  - configuració del kernel de Linux
  - suport de mòduls
- Suport hardware
  - sincronització
  - comptadors hardware

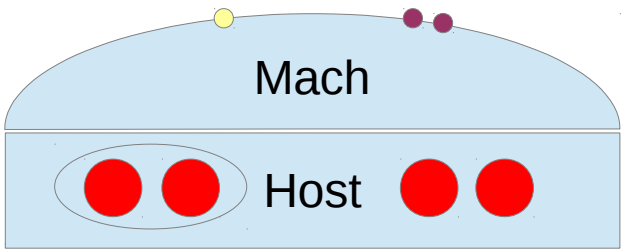
Interfície de Mach

- Abstraccions
  - Host
  - Processor / Processor set
  - Device
  - Task
  - Thread
  - VM (virtual memory)
  - Memory object
  - Port / Port set
  - Message



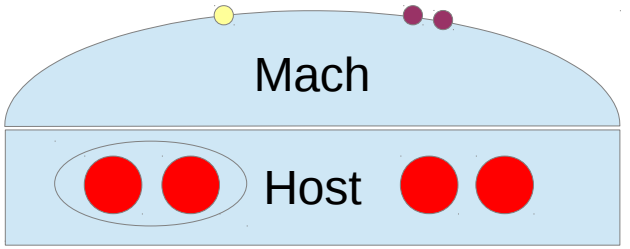
Host interface

- Obtenció d'informació
  - Bàsica: processadors, tipus i memoria disponible
  - Càrrega instantània (uptime, top)
  - Planificació: timeout, quantum mínims (en ms.)
  - Versió del kernel
  - Data i hora



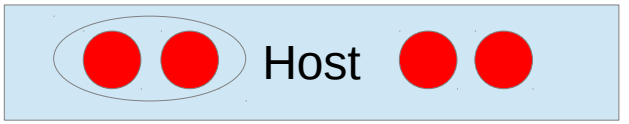
Host interface

- Canvi de la informació
  - Ajustar data i hora
- Interactuar amb el host
  - Reboot
- [Aconseguir els ports privilegiats del host]
  - ofert pels servidors de sistema(?)
  - int get_privileged_ports(
    &host_privileged_port,
    &device_privileged_port
    );
    - Nomes accessible a root



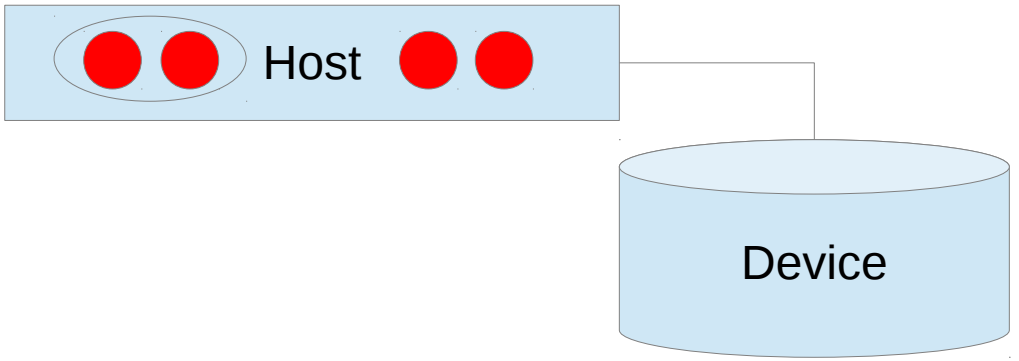
Processor interface

- Aconseguir informació
  - Tipus, si està funcionant, slot, master
  - Processor set al qual està vinculat
- Parar i engegar processadors
- Obtenir la llista de processor sets
- Obtenir accés a processor sets per nom
- Assignar i desassignar processadors a processor sets
- Crear i destruir processor sets
- Obtenir informació del processor set
  - Tasks associades
  - Threads associats



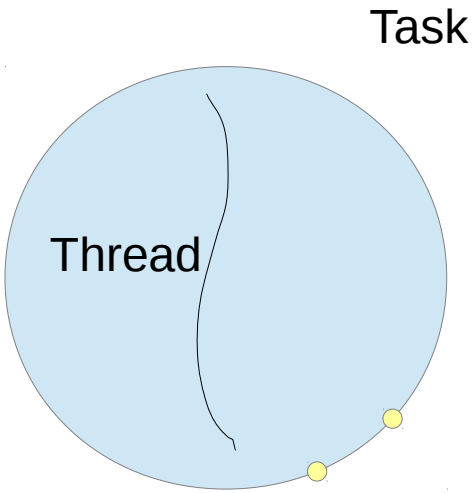
Device interface

- Open / close / read / write / map
- Asynchronous open / read / write
- Status
- Filter



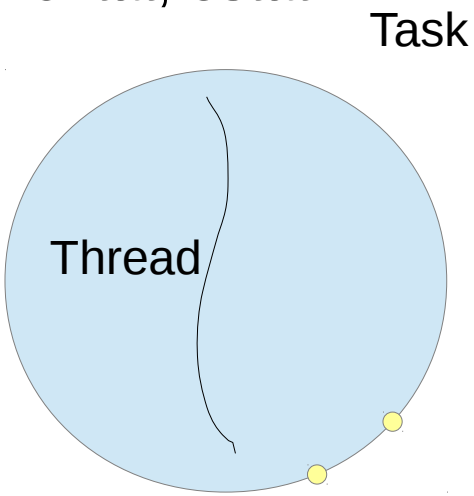
Task interface

- Creació / destrucció
- Aturar / continuar
- Canviar prioritat
- Aconseguir informació
  - comptador d'aturades, prioritat, mides de memòria
  - temps d'execució per threads vius i acabats
  - llista de threads
- mach_task_self (3.0) / task_self (2.5)
- get / set special ports
  - bootstrap, exception, kernel



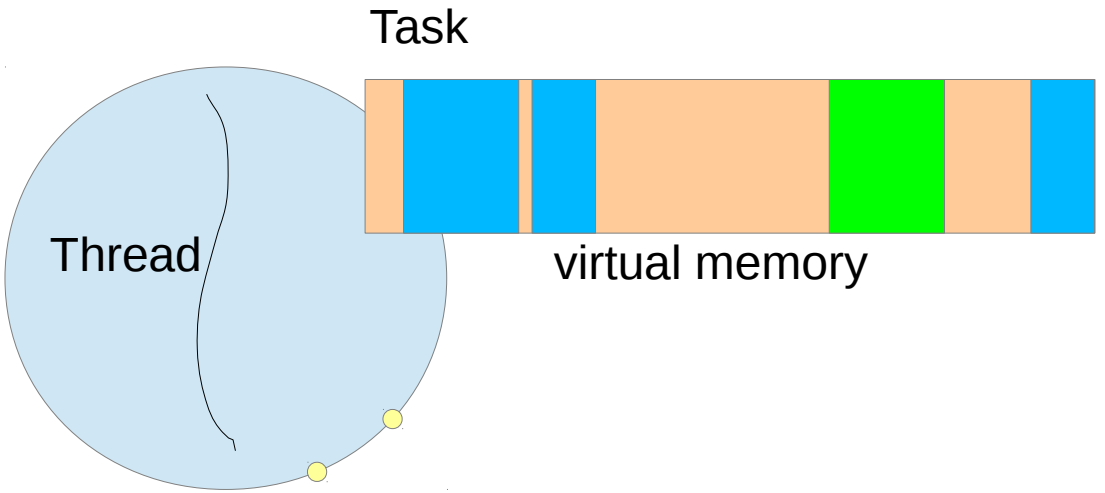
Thread interface

- Creació / destrucció / consultar/assignar registres
- Aturar / continuar / canvi de context
- Canviar prioritat, política de planificació i assignació a processadors
- Aconseguir informació
  - comptador d'aturades, política de planificació, prioritat, estat
  - temps d'execució, ús de cpu
- mach_thread_self (3.0) / thread_self (2.5)
- get / set special ports
  - exception, kernel



Virtual Memory interface

- Demanar i alliberar memòria anònima
- Còpia entre regions de memòria
- Mapejar **memory objects**
- Fixar la memòria virtual a la física
- Obtenir informació sobre les regions de memòria



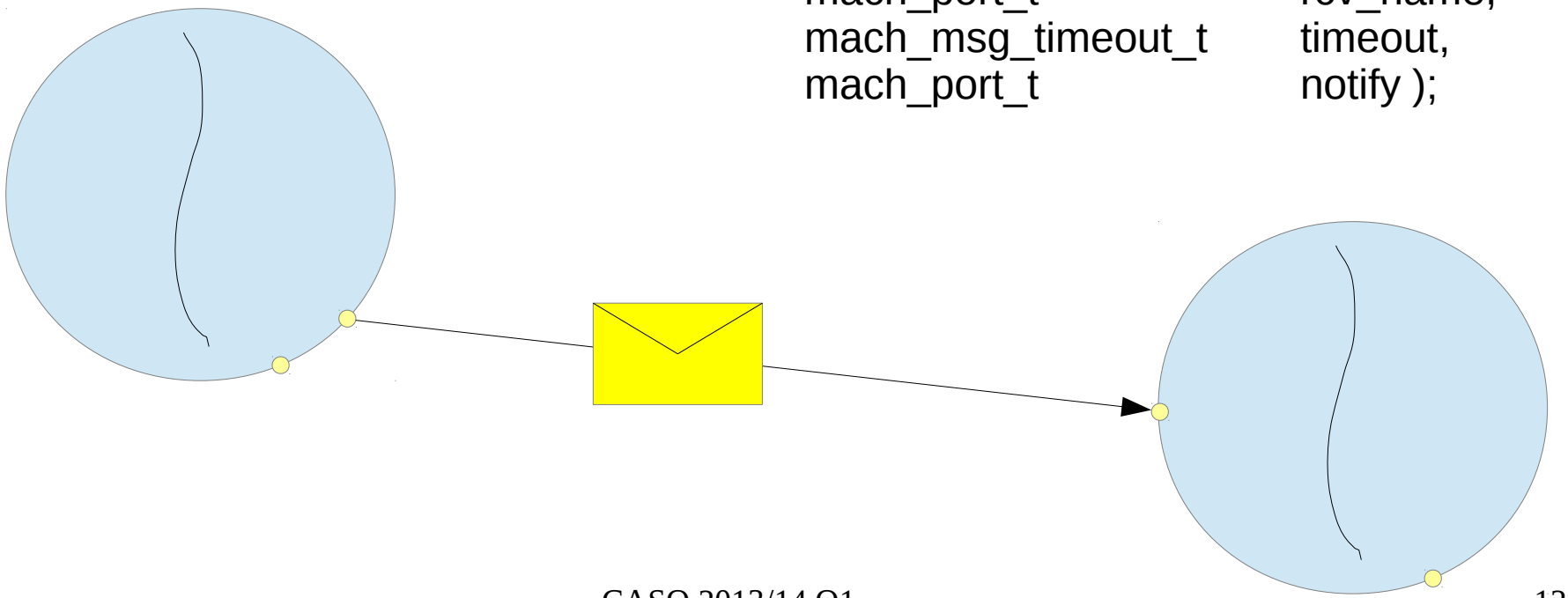
Port interface

- Creació / destrucció de ports i port sets
- Canvi de drets sobre ports
  - Enviar / rebre
- Informació sobre un port
  - Comptador de referències
  - Estat
- Creació /destrucció de port sets
- Moure ports a port sets

Message interface

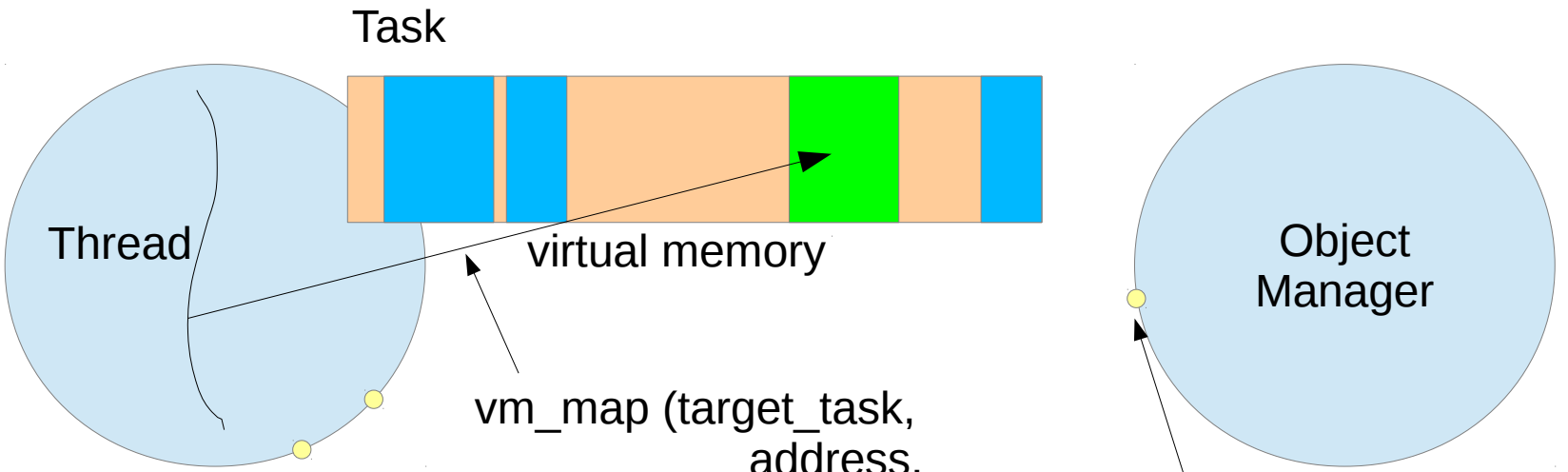
- Enviar i rebre missatges a ports

```
mach_msg_return_t mach_msg
(mach_msg_header_t*      msg,
 mach_msg_option_t       option,
 mach_msg_size_t         send_size,
 mach_msg_size_t         rcv_size,
 mach_port_t             rcv_name,
 mach_msg_timeout_t      timeout,
 mach_port_t             notify );
```



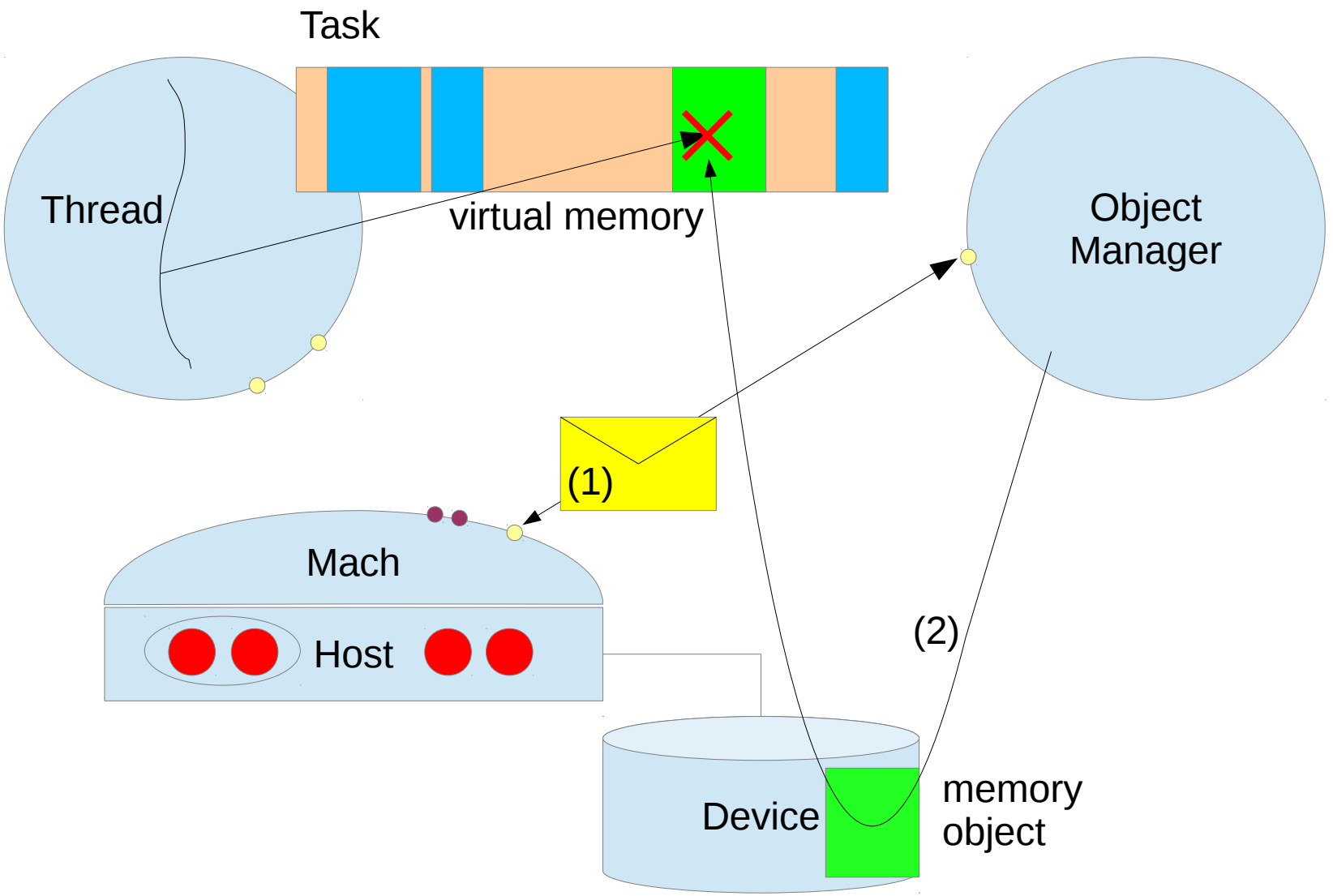
Memory Object interface

- Permet que processos d'usuari s'encarreguin de la gestió de la memòria virtual




```
vm_map (target_task,
        address,
        size,
        alignment_mask,
        anywhere,
        memory_object,
        offset,
        copy_or_map,
        cur_protection,
        max_protection,
        inheritance
);
```

Memory Object interface



Memory Object interface

- (1)
  - memory_object_init /
    memory_object_ready
  - memory_object_data_request / → pagein
    memory_object_data_supply → dades
    memory_object_data_unavailable → zeros
    memory_object_data_error → segfault
  - memory_object_data_return → pageout
  - memory_object_data_unlock → permetre accés

Memory Object interface

- (2)
  - vm_allocate / vm_deallocate
  - device_read / device_write

Índex

- Interfície de Mach
- Eines de desenvolupament
  - gcc/binutils
  - configuració del kernel de Linux
  - suport de mòduls
- Suport hardware
  - sincronització
  - comptadors hardware

Linux: estructura del codi font

```
COPYING          MAINTAINERS     block/     include/  mm/         sound/
CREDITS          Makefile        crypto/    init/     net/        tools/
Documentation/   README          drivers/   ipc/      samples/    usr/
Kbuild           REPORTING-BUGS  firmware/  kernel/   scripts/    virt/
Kconfig          arch/           fs/        lib/      security/
```

- Habitualment a /usr/src/linux
- Escrit en C
  - gcc/binutils
  - make
  - Kconfig

Eines de desenvolupament

- Sistema i aplicacions
  - GCC (GNU Compiler Collection)
    - gcc-4.7.1
  - GNU Binutils
    - binutils-2.22
  - Make
- Aplicacions
  - Autotools
    - autoconf, automake, libtool, gettext

http://gcc.gnu.org/
http://www.gnu.org/software/binutils/
http://www.gnu.org/software/make/

GCC now uses C++ as its implementation language [2012-08-14]

Binutils

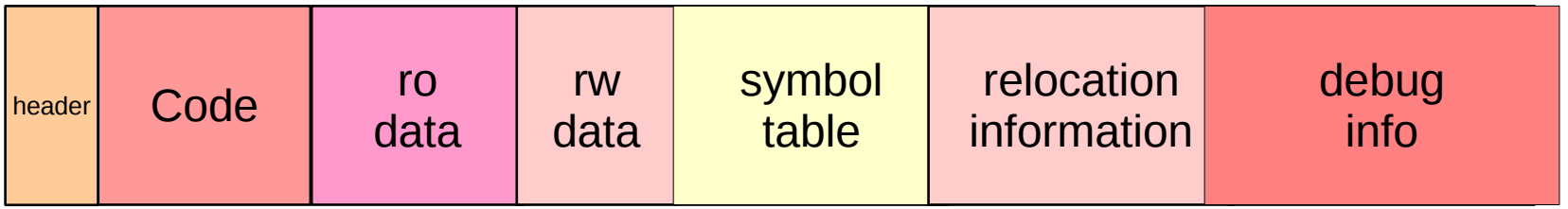
- ld / gold — linker
- as — assembler
- ar / ranlib / dlltool — library management
- gprof — profiler
- nm / objdump / readelf — object viewers
- size / strings — simple viewers

Binutils

- objcopy object management
- strip eliminate symbols/debug
- addr2line converts addresses to line nums
- c++filt c++ symbol demangling
- nlmconv / windmc / windres
  - Transformation tools for NetWare and Windows

Format dels executables: ELF

- ELF (Executable and Linkable Format)
  - Codi
  - Dades (només lectura / lectura-escriptura)
  - Símbols i les seves adreces
  - Reubicacions
  - Informació de debug
    - Números de línia
    - Tipus, estructures, variables...
    - ...



Exemples

- strings -a <fitxer> aplicable a qualsevol fitxer
  - Dóna els "strings" que troba dins el fitxer
    - Definits com a "3 o més caràcters imprimibles"
- objdump -d <fitxer.o | executable>
  - Desassembla el codi contingut en el fitxer
- objdump <-t | -T> < fitxer.o | executable>
  - Dóna els símbols locals al fitxer (-t) / públics per linkar amb altres fitxers (-T)

Estructura del kernel

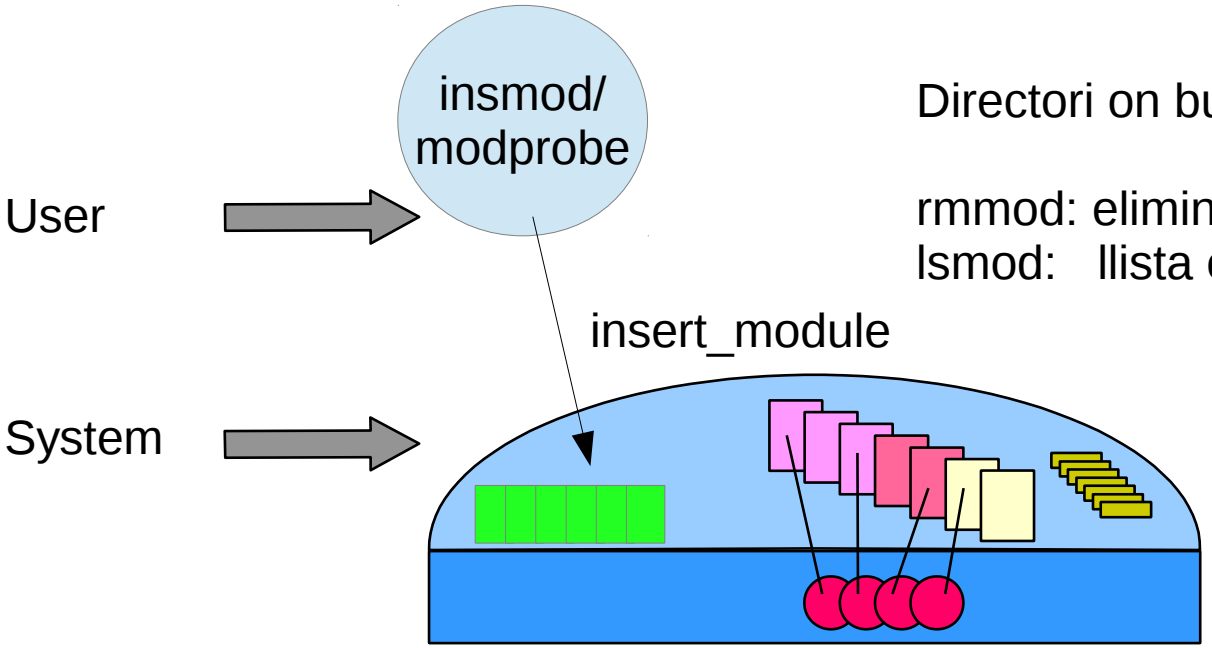
- Monolític, amb **mòduls**
  - Parts del kernel de Linux es poden incorporar dinàmicament

insmod <mòdul>.ko: inserta el mòdul donat

modprobe <mòdul>: busca el mòdul per nom i l'inserta

Directori on buscar: /lib/modules/<kernelversion>

rmmod: elimina un mòdul de l'interior del kernel
lsmod: llista els mòduls carregats

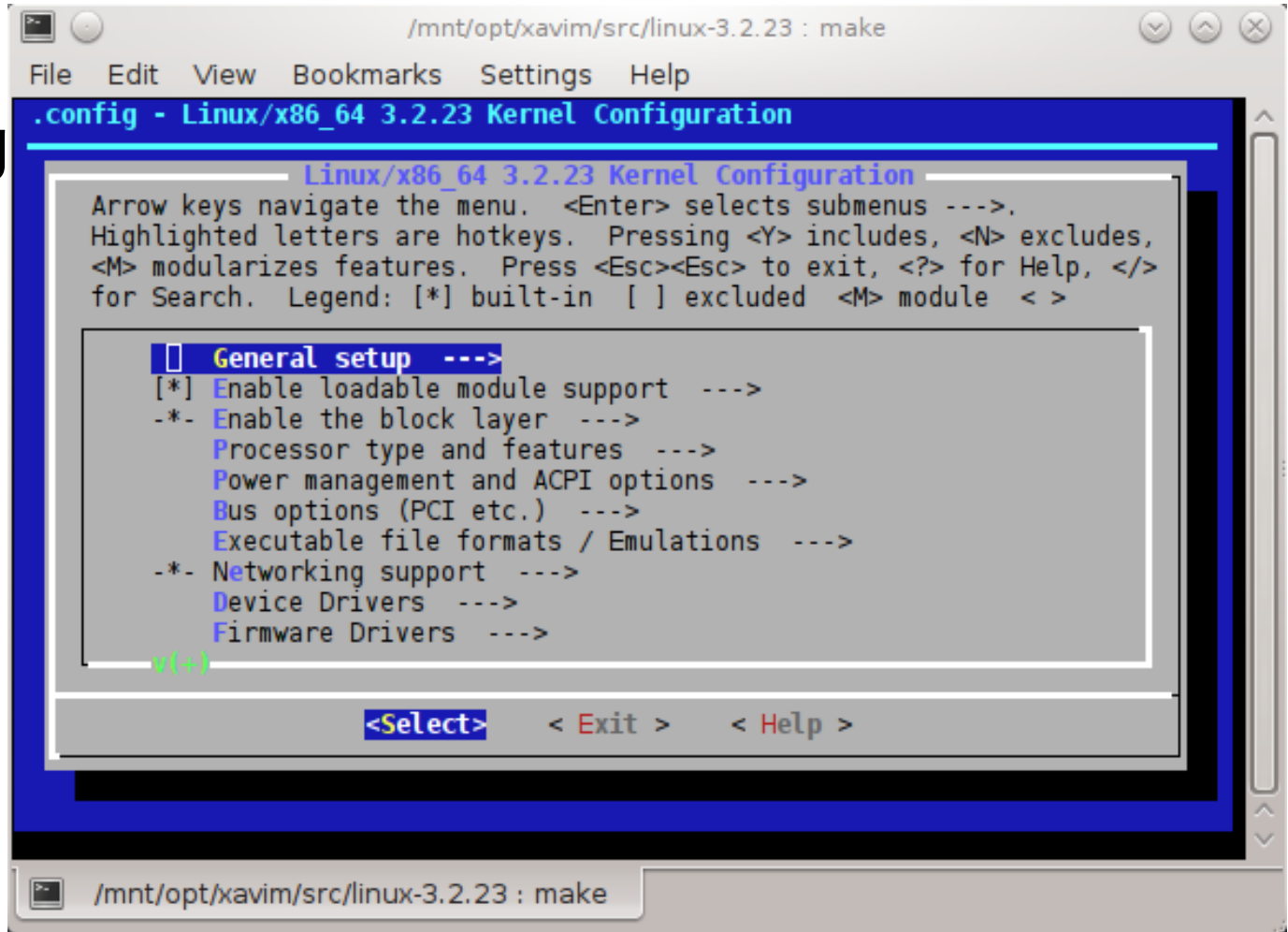


Estructura del kernel

- Mòduls, permeten que el kernel sigui més petit
- Habitualment usats en:
  - Gestors de dispositiu
    - Xarxa, so, bateria, dispositius USB...
  - Sistemes de fitxers
    - vfat, ext2/3/4, reiserfs, ntfs, quota...
  - Protocols de xarxa (ipv6)
  - Gestió del consum (cpufreq)
  - ...

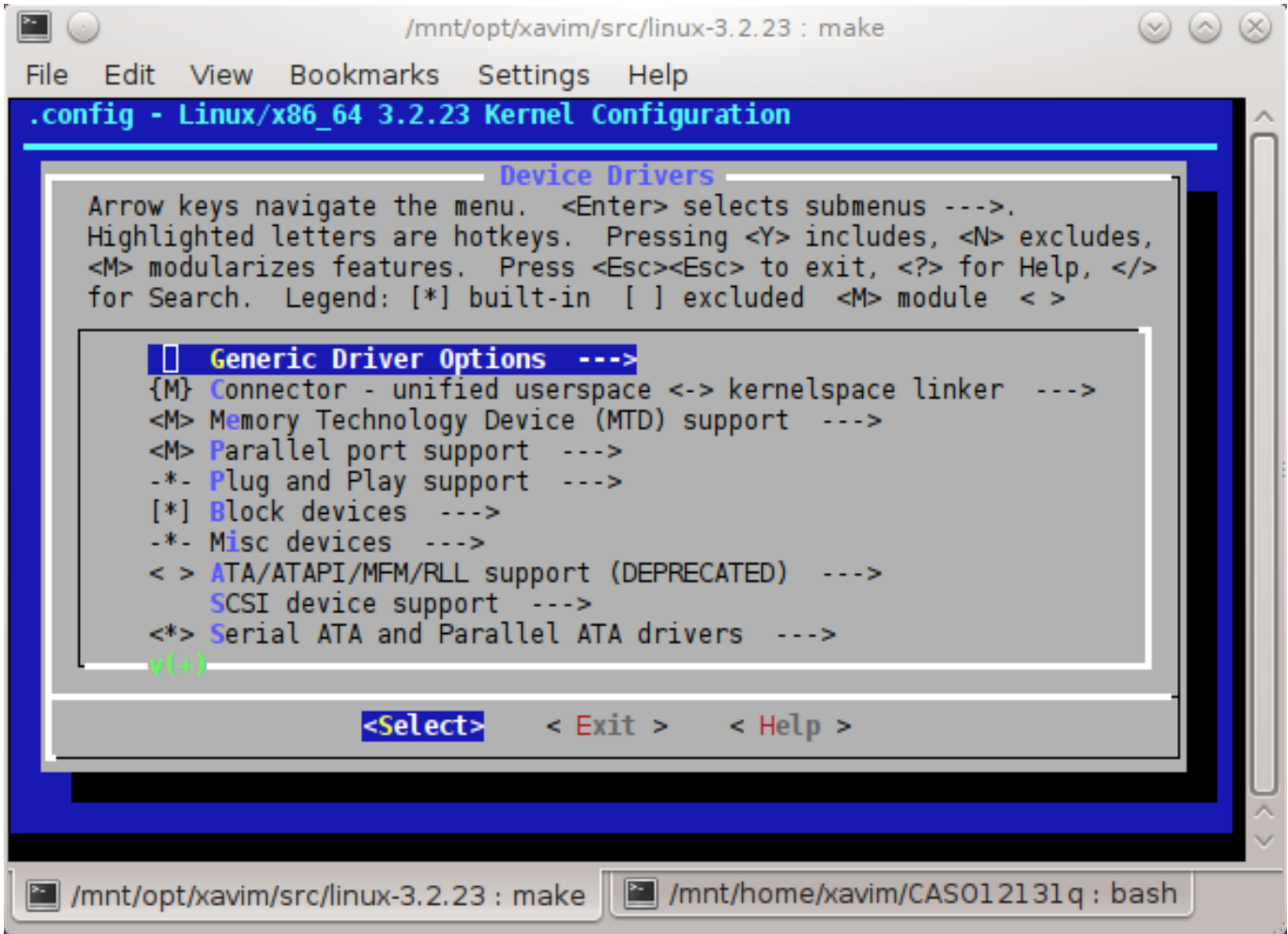
Configuració del kernel

- Des de /usr/src/linux
  - make config
  - **make menuconfig**
    - Centenars d'opcions
    - Configuració actual:
      - .config



Configuració del kernel

- Selecció dels components
  - [*] - seleccionat, no pot ser mòdul
  - <M> – com a mòdul
  - -*- inclòs, seleccionat per una altra opció
  - {M}, idem, com a mòdul



Compilació de mòduls

- Incorporats al kernel (in-tree)
  - l'habitual, realitzada amb el codi del mòdul **dins** dels fonts de Linux (/usr/src/linux)
- Externs (out-of-tree)
  - amb el codi del mòdul **fora** dels fonts de Linux
  - interessant per iniciar el desenvolupament
  - o per a que "3rd parties" puguin distribuir els seus mòduls

Moduls externs

- Requeriments
  - Tenir un kernel precompilat
    - habitualment a /usr/src/linux
    - amb els fitxers de configuració i d'include disponibles
    - amb el suport de mòduls activat
- Usa un mecanisme específic: kbuild i make

linux/Documentation/kbuild/modules.txt

http://tldp.org/HOWTO/pdf/Module-HOWTO.pdf

Moduls externs

- Comanda per compilar el mòdul
  - make -C /usr/src/linux/ M=$PWD [modules]
- Comanda per instal·lar el mòdul
  - make -C /usr/src/linux M=$PWD modules_install

Fitxer **Makefile**

```
ifneq ($(KERNELRELEASE),)     # for kbuild

obj-m := mymodule.o
mymodule-y := interface.o implementation.o

else                                  # normal Makefile

KDIR ?= /lib/modules/`uname -r`/build
default:
	$(MAKE) -C $(KDIR) M=$$PWD

endif
```

Índex

- Interfície de Mach
- Eines de desenvolupament
  - gcc/binutils
  - configuració del kernel de Linux
  - suport de mòduls
- Suport hardware
  - sincronització
  - comptadors hardware

Suport hardware

- Cap als spin locks
  - Requereixen suport d'instruccions atòmiques
  - Versió ideal, però **no funcional**:
    - L'execució de l'entrada a la regió crítica no és atòmica

```
int lock;

…
```

Flux 1

```
while (lock==1) ;   //spin
lock = 1;

   // regió crítica de codi



lock = 0;
```

Flux 2

```
while (lock==1) ;   //spin
lock = 1;

   // regió crítica de codi



lock = 0;
```

Suport hardware

- Cap als spin locks
  - Suport del compilador
    - Atribut "volatile", indica que un altre flux pot estar accedint a la mateixa variable al mateix temps
    - Encara **no funcional**

volatile int lock;

… **Flux 1**

```
while (lock==1) ;    //spin
lock = 1;

   // regió crítica de codi



lock = 0;
```

Flux 2

```
while (lock==1) ;   //spin
lock = 1;

   // regió crítica de codi



lock = 0;
```

Suport hardware

- Suport per spin-locks
  - A través d'intríseques del compilador (gcc)

volatile int lock;

…

```
while (__sync_lock_test_and_set (&lock, 1)==1) ;    //spin

   // regió crítica de codi



lock = 0;
```

Suport hardware

- Suport per spin-locks
  - Què genera el compilador?

```
bash-4.2$ objdump -d spin.o

0000000000000000 <f>:
   0:   55                      push   %rbp
   1:   48 89 e5                mov    %rsp,%rbp
   4:   90                      nop
   5:   b8 01 00 00 00          mov    $0x1,%eax
   a:   87 05 00 00 00 00       xchg   %eax,0x0(%rip)        # 10
<f+0x10>
  10:   83 f8 01                cmp    $0x1,%eax
  13:   74 f0                   je     5 <f+0x5>
  15:   bf 00 00 00 00          mov    $0x0,%edi
  1a:   e8 00 00 00 00          callq  1f <f+0x1f>           printf ("Hola\n");
  1f:   c7 05 00 00 00 00 00    movl   $0x0,0x0(%rip)        # 29
<f+0x29>
  26:   00 00 00
  29:   5d                      pop    %rbp
  2a:   c3                      retq
```

Suport hardware

- Evitar la sobrecàrrega d'instruccions en els multicore d'Intel
  - Instrucció PAUSE
    - Fa que només hi hagi un "load" en vol
    - Facilita la invalidació (squash) de les instruccions quan el processador detecta una invalidació de la línia de cache

```
while (sync_var != value)
    asm __volatile__ (“pause”);
```

- Evitar la sobrecàrrega del bus, per la transacció atòmica
  - Test, test-and-set

```
while (__sync_lock_test_and_set (&sync_var, BUSY)==BUSY)
  while (sync_var==BUSY) asm __volatile__ (“pause”);
```

Suport hardware

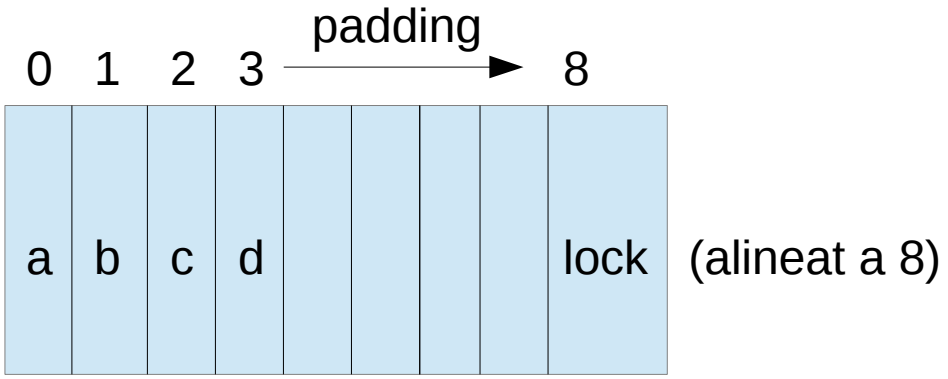
- Que genera el compilador?

```
0000000000000000 <f>:
   0:   ba 01 00 00 00          mov    $0x1,%edx
   5:   0f 1f 00                nopl   (%rax)
   8:   89 d0                   mov    %edx,%eax
   a:   87 05 00 00 00 00       xchg   %eax,0x0(%rip)        # 10 <f+0x10>
  10:   83 f8 01                cmp    $0x1,%eax
  13:   75 12                   jne    27 <f+0x27>
  15:   0f 1f 00                nopl   (%rax)
  18:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # 1e <f+0x1e>
  1e:   83 f8 01                cmp    $0x1,%eax
  21:   75 e5                   jne    8 <f+0x8>
  23:   f3 90                   pause
  25:   eb f1                   jmp    18 <f+0x18>
  27:   48 83 ec 08             sub    $0x8,%rsp
  2b:   bf 00 00 00 00          mov    $0x0,%edi
  30:   e8 00 00 00 00          callq  35 <f+0x35>
  35:   c7 05 00 00 00 00 00    movl   $0x0,0x0(%rip)        # 3f <f+0x3f>
  3c:   00 00 00
  3f:   48 83 c4 08             add    $0x8,%rsp
  43:   c3                      retq
```

Suport hardware

- Deixar espai ("padding") entre variables de sincronització
  - Per evitar col·lisions a la cache
- Alinear variables almenys a la seva pròpia mida
  - Alinear implica "deixar padding, si cal"

volatile int lock __attribute__ ((aligned(128))); // alineat a 128 bytes



Suport hardware

- Suport del compilador
  - Funcions "intrinsic" per accedir a memòria
    - Inclouen una "barrera de memòria", memory barrier
      - El compilador no pot optimizar codi i moure les altres instruccions al voltant d'aquestes
      - Mantenen la semàntica del codi

```
TYPE __sync_fetch_and_add (TYPE *ptr, TYPE value, ...)     +sub, or, and, xor, nand
TYPE __sync_add_and_fetch (TYPE *ptr, TYPE value, ...)     + ...
TYPE __sync_bool_compare_and_swap (TYPE *ptr, TYPE oldval, TYPE newval, ...)
TYPE __sync_val_compare_and_swap (TYPE *ptr, TYPE oldval, TYPE newval, ...)
TYPE __sync_lock_test_and_set (TYPE *ptr, TYPE value, ...)
void __sync_lock_release (TYPE *ptr, ...)
__sync_synchronize (...)
```

Suport hardware

- __sync_fetch_and_add (&val, 1)
  - lock addl $0x1,2099117(%rip) # <val>
- __sync_lock_test_and_set (&lock, BUSY)
  - xchg %eax, 2099015(%rip) # <lock>
- __sync_synchronize ()
  - No és necessària en Intel x86, x86_64
  - Necessària en altres arquitectures
    - IBM Power...

Suport hardware

- Comptadors hardware
  - Permeten comptar events que passen durant l'execució
  - Per codi d'usuari, sistema, excepció
- Limitations
  - Número petit de registres per comptar
    - Multiplexar events en els registres disponibles -> SO
  - Registres "petits" (32bits)
    - Capturar excepcions en overflow -> SO (signal)
  - Depenents de l'arquitectura i del processador
    - Processadors de la mateixa família tenen comptadors diferents

Suport hardware

- Events interessants
  - Cycles
  - Issued, graduated instructions
  - Issued, graduated loads/stores
  - Level N cache misses
  - TLB hits / misses
  - ALU/FPU progress cycles
  - Number of specific FP instructions issued, graduated
  - External intervention hits
  - External invalidations
  - and many more...
- Deriving other metrics
  - Memory bandwidth
  - MFlops
  - IPC
  - ...

Suport hardware

- Incorporar els comptadors en el context dels processos/fluxos
- Perfctr
  - Linux, incorporat en el sistema
  - Part de sistema, manteniment dels comptadors
  - Part d'usuari, eines per treure els comptadors i analitzar el seu significat
- PAPI: llibreria de suport per extreure la informació

Suport hardware

- Informació del procés/flux
  - a través de crides a sistema
- Informació del processador
  - [Quin procés està executant actualment?]
  - Quan de temps està dedicant a executar codi
    - d'usuari?
    - de sistema?
    - en interrupció?

Exemple: interfície de PAPI

- PAPI_start_counters (events, number)
- PAPI_read_counters (values, number)
- PAPI_accum_counters (values, number)
- PAPI_stop_counters (values, number)
- PAPI_flops (...)

Per a la setmana vinent

- Documentar-se
  - buscar informació sobre virtualització
    - Linux containers
    - VirtualBox
    - Bochs
    - QEMU
    - KVM
    - Hyper-V
- Entregueu una taula amb les següents columnes:
  - Web (http://...)
  - Programari lliure (sí/no)
  - Arquitectures on es pot executar
  - Emulació de diferents arquitectures (sí – quines / no)
  - Sistemes operatius on es pot executar (quins)
  - Sistemes operatius que pot emular (quins)