Fonaments dels Sistemes Operatius (FSO)

Departament d'Informàtica de Sistemes i Computadors (DISCA)
Universitat Politècnica de València

Bloc Temàtic 1: Introducció

Unitat Temàtica 2:
Concepte de Crida al Sistema

fSO

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DISCA

Concepte de Crida al Sistema

- **Objectius**
  - Proporcionar una visió global del **funcionament del computador**, destacant els aspectes que afecten de forma directa al sistema operatiu.
  - Presentar el concepte de **crida al sistema** com el mecanisme necessari per a obtenir els serveis del sistema operatiu.
  - Descriure els **serveis** que el sistema operatiu proporciona als usuaris i als processos.

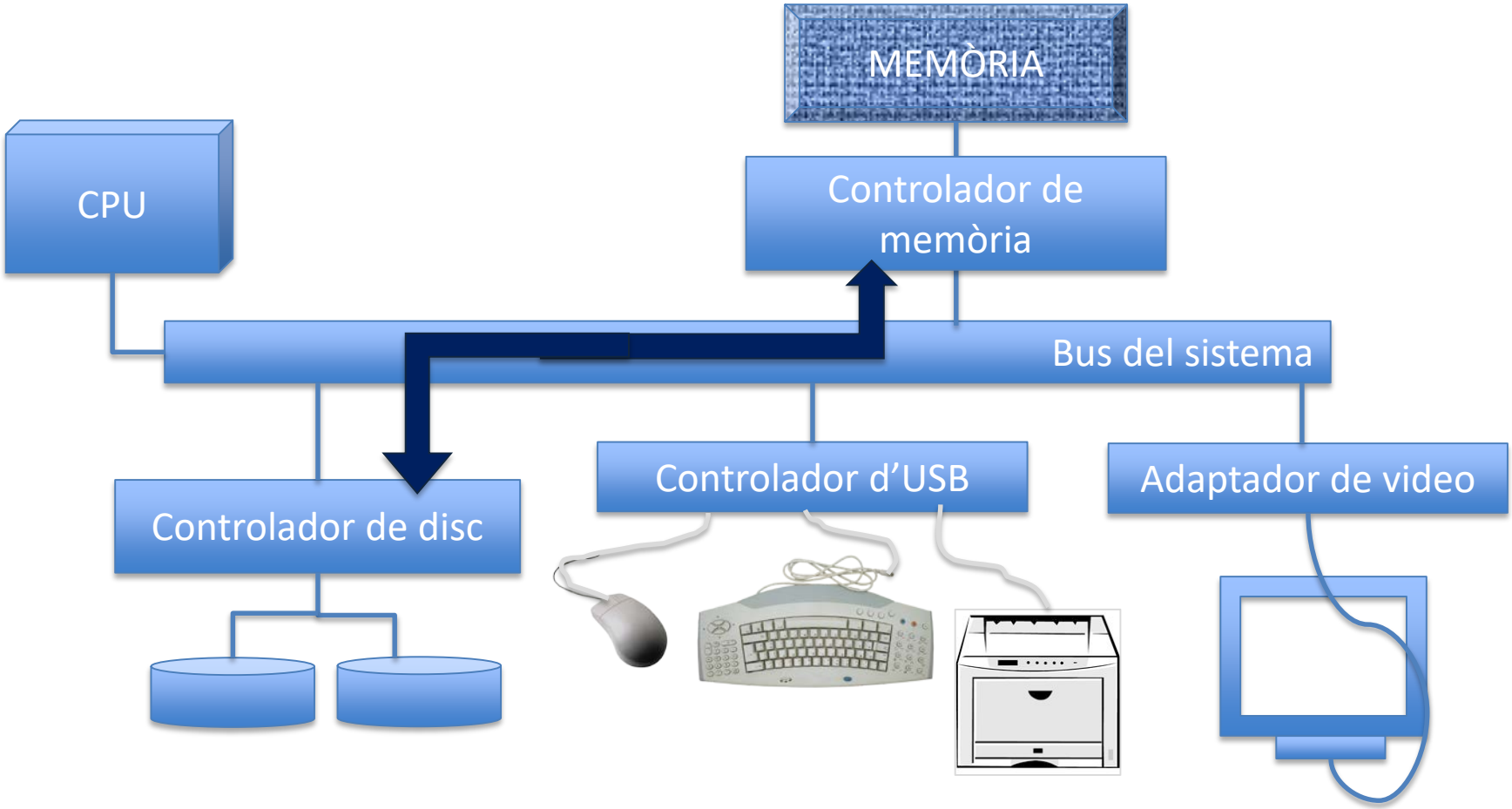
- **Bibliografia**
  - A. Silberschatz, P. B. Galvin. “Sistemas Operativos”. 7ª ed. Capítols. 1 i 2

- **Arquitectura hardware del computador**
- Interrupcions
- Modes d'execució
- Crides al sistema
- Utilitats del sistema.

Nomenclatura:	
DMA	Accés directe a memòria (*direct memory access*)
SO	Sistema Operatiu
USB	Bus universal en sèrie (Universal Serial Bus)
INT	Interrupció
POSIX	Portable Operating System Interface

- Concurrència entre entrada/eixida i CPU
  - Els dispositius d'**E/S són més lents que els processadors**
    - Exemple: el temps que cal per a accedir a una informació emmagatzemada en un disc
  - Un processador modern és capaç d'executar **milers de milions d'instruccions màquina** en el temps que se tarda a accedir al disc.
  - Mentre s'hi fan operacions d'E/S el processador ha de poder executar instruccions útils en lloc d'un programa d'espera→ **Concurrència entre CPU i E/S**

- Funcionament d'un sistema informàtic



- Manejador versus Controlador

Sistema operatiu

Manejador de dispositiu (Device Driver)
-Element del sistema operatiu
-Software amb capacitat
-Per a programar els controladors
-Proporciona una interfície amigable per a l'ús del controlador

Hardware dels dispositius d'E/S

Controlador de dispositiu
-Component hardware
-Registres de control
-Buffer d'emmagatzema-ment
-Registre d'estat
-Capacitat de DMA

Contingut

- Arquitectura hardware del computador
- **Interrupcions**
- Modes d'execució
- Crides al sistema
- Utilitats del sistema.

Interrupcions

- Un sistema operatiu és un **programa dirigit per esdeveniments.**
- Aquests esdeveniments són les i**nterrupcions hardware,** les **interrupcions software** i les **excepcions.**
- El SO actua com a un programa servidor a l'espera que se li encomane treball mitjançant interrupcions
- Els processos i els dispositius d'E/S soliciten serveis al SO

Interrupcions

- ¿Qui genera una sol·licitud d'interrupció?
- ¿Quan es genera una sol·licitud d'interrupció?

INTERRUPCIÓ

Interrupció d'E/S:
Generada per controladors de dispositius d'E/S

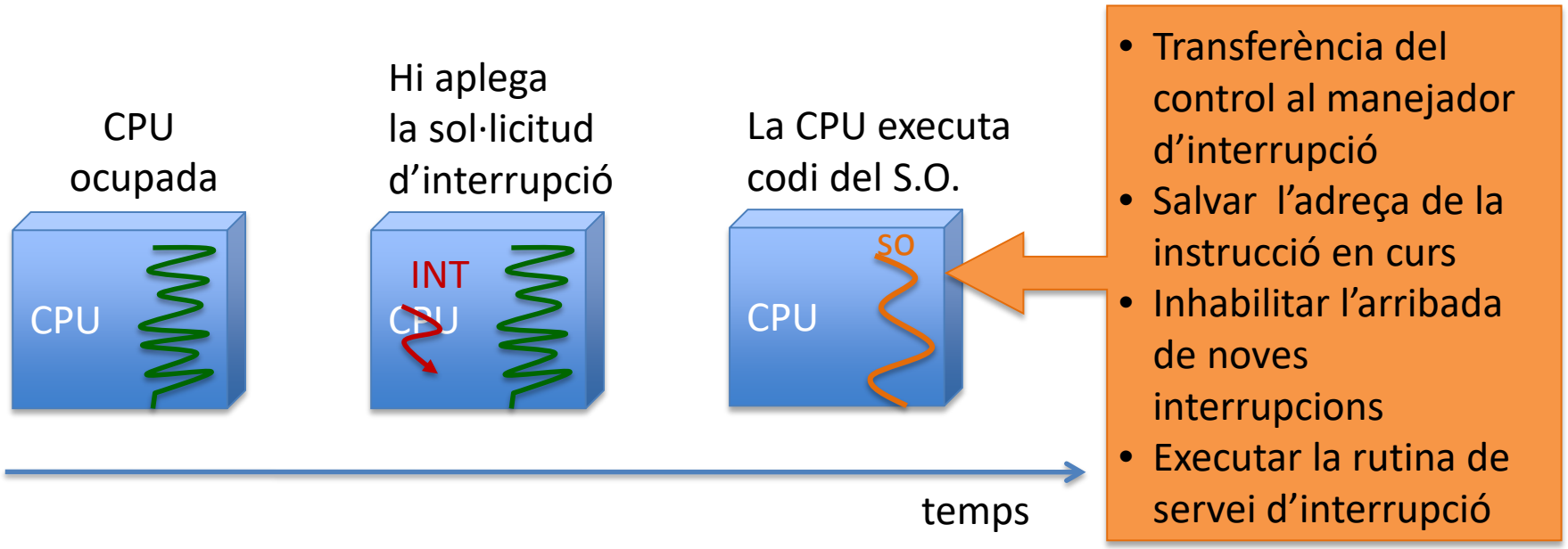
Interrupció de Rellotge:
El s.o. entra a executar-se cada cert intèrval

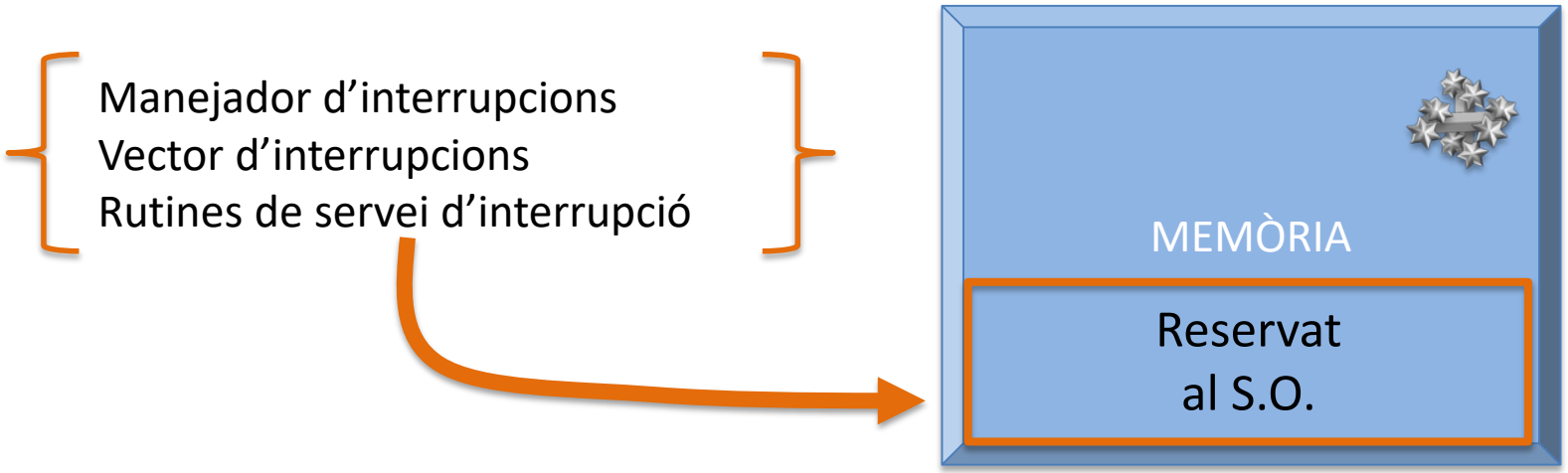
Excepció hardware:
Error de paritat en memòria, tall de corrent...

Interrupcions Traps:
Utilitzades pels programes per a sol·licitar serveis al S.O.

Excepcions software:
Es generan quan en els programes hi ha una divisió per zero, desbordament d'operacions aritmètiques, adreçament d'una posició de memòria prohibida...

- Mecanisme d'interrupció





Contingut

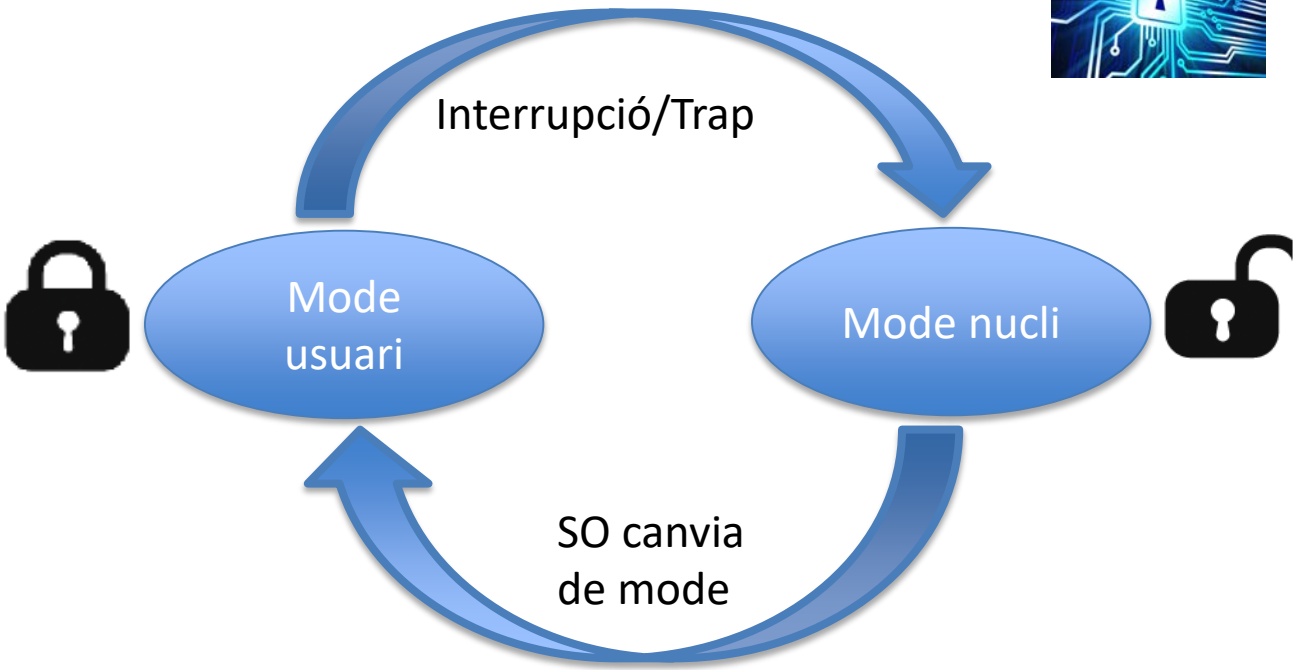
- Arquitectura hardware del computador
- Interrupcions
- **Modes d'execució**
- Crides al sistema
- Utilitats del sistema.

Modes d'execució

- Modes d'execució del processador
  - Els processadors presenten **dos o més modes d'execució**
    - Els modes d'execució s'incloen per donar **suport al sistema operatiu**
    - Els processos que s'executen simultàniament comparteixen recursos de la màquina → **necessidad de protecció**
  - Protecció de l'accés al hardware per a impedir que els programes d'usuari
    - accedisquen a la **memòria** del sistema lliurement
    - monopolitzen l'ús **de la CPU**
    - accedisquen a certs **registres del sistema**
    - accedisquen directament als **dispositius d'E/S**
  - Els processadors actuals disposen d'un **bit de mode**, per a indicar el mode actual, ***nucli(0) – usuari (1)***

Modes d'execució

- Dos modes d'execució
  - **Mode usuari o regular**: té restringit el joc d'instruccions
  - **Mode nucli o privilegiat**: permet executar tot tipus d'operacions hardware, accedir a memoria i als dispositius d'E/S
- Mode dual d'operació



Modes d'execució

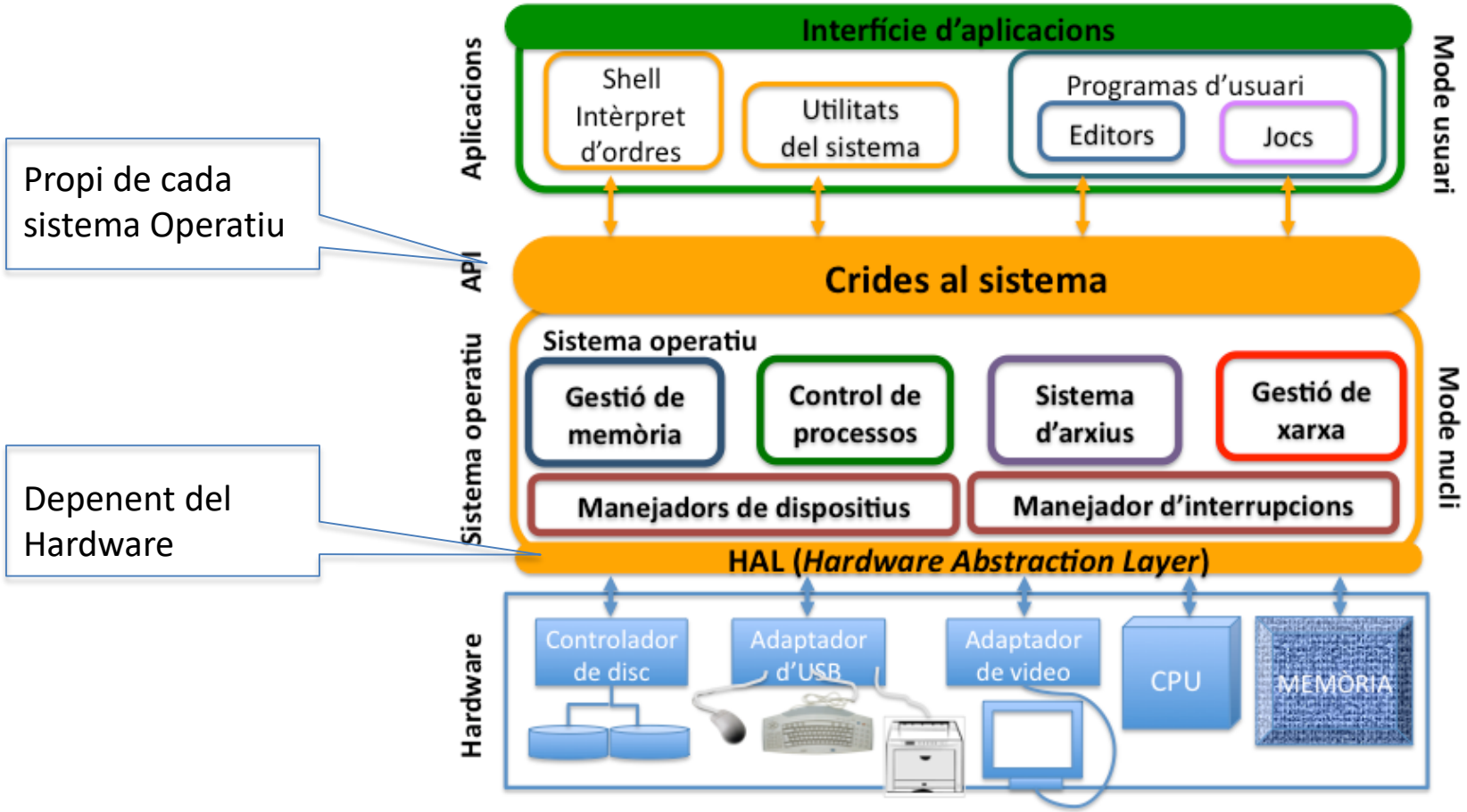
- Instruccions privilegiades
  - Són aquelles **instruccions disponibles en mode nucl**i que no ho estan en mode usuari.
  - Les instruccions privilegiades estan associades principalment a tres tipus de protecció
    - Protecció de l'entrada/sortida
    - Protecció de la memòria
    - Protecció del processador

Contingut

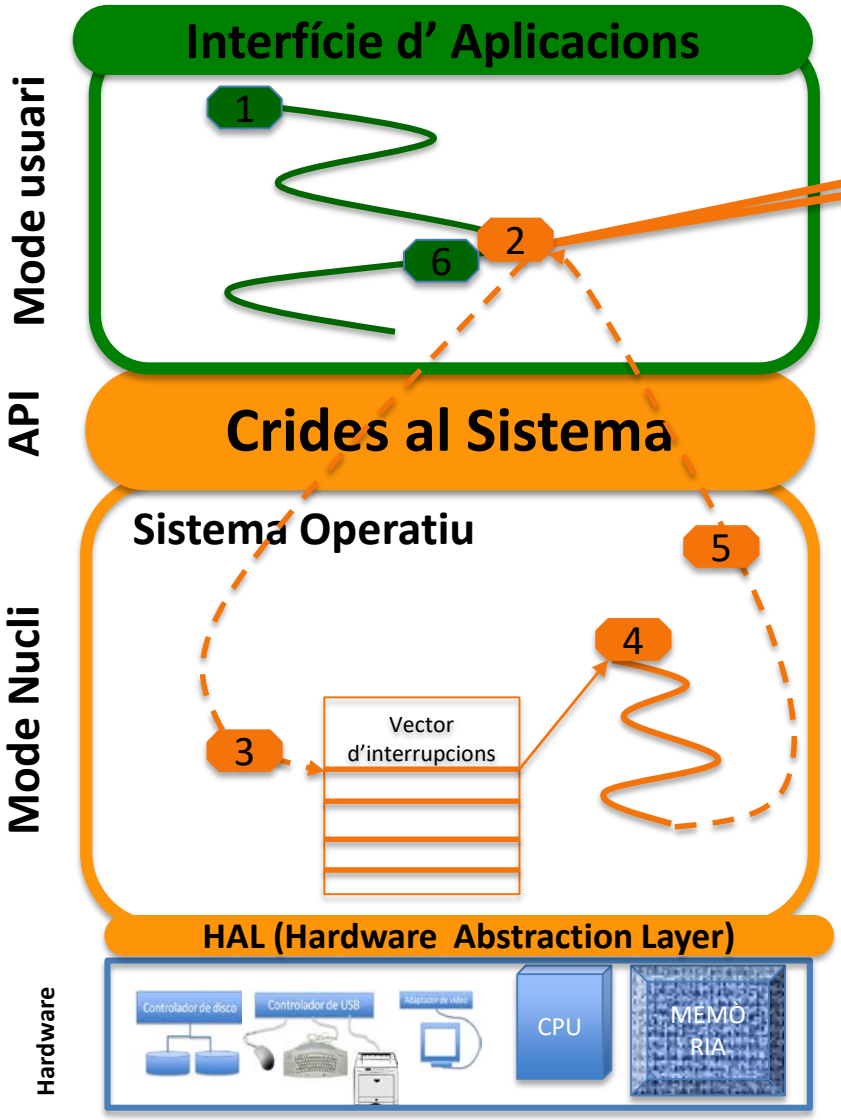
- Arquitectura hardware del computador
- Interrupcions
- Modes d'execució
- **Crides al sistema**
- Utilitats del sistema.

Crides al sistema

- Mecanismes per a **sol·licitar serveis** al sistema operatiu.
- **Interfície** proporcionada pel S.O. per a accedir als recursos hardware de la máquina.
- Utilitat en forma de **funcions de biblioteca** per a accedir a recursos gestionats pel sistema operatiu.



- Sol·licitud de servei al Sistema Operatiu



Mode usuari

1. Programa executant-se
2. Crida al sistema (*Trap* o Interrupció software)

Mode nucli

3. Identificació del servei sol·licitat
4. Execució del servei sol·licitat
5. Dades sol·licitades, resultat del servei

Mode usuari

6. Següent instrucció del programa

¡¡ATENCIÓ!!
¿Compatibilitat entre SO? NO
Cada sistema operatiu té les seues pròpies crides

Crides al sistema

- POSIX: Exemple de crides
- Estàndar POSIX (Portable Operating System Interface)

	Processos
fork	Creació d'un procés fill
exit	Acabament del procés en execució
wait	Espera l'acabament d'un procés
exec	Canvia la imatge de memòria per la d'un executable (executa progr
getpid	Obté atributs d'un procés
setsid	Modifica atributs d'un procés

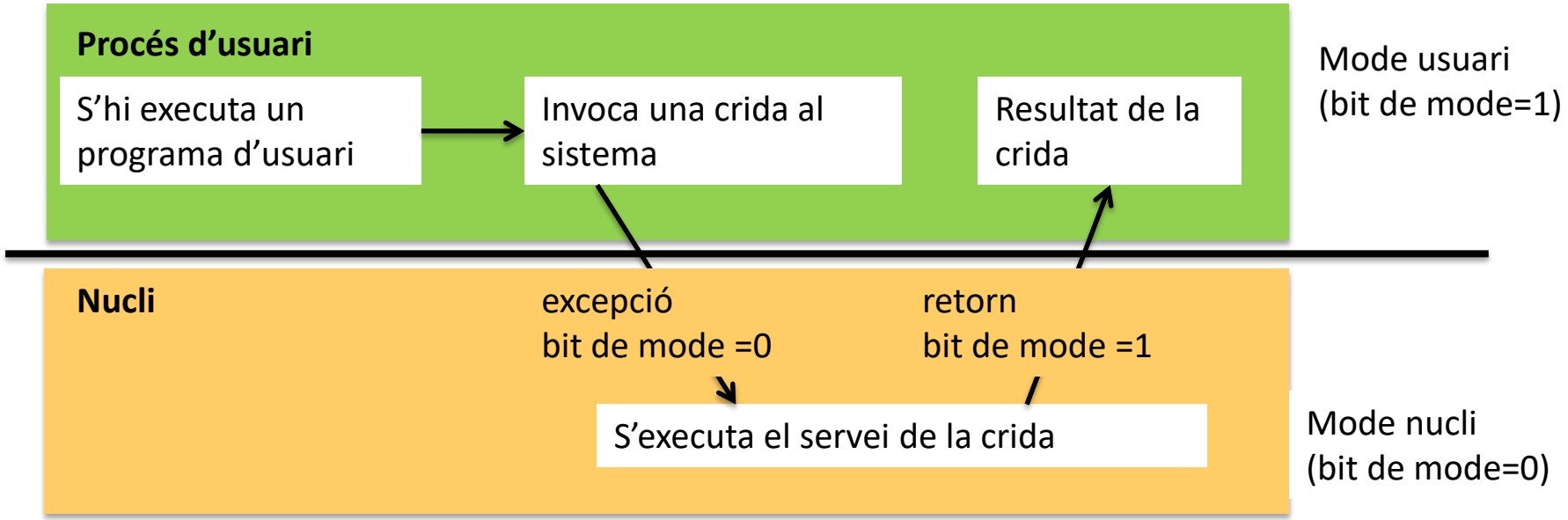
	Directoris
mkdir	Crea directori
rmdir	Elimina un directori buit
opendir	Obri directoris
readdir	Retorna la següent entrada d'un directori
closedir	Tanca un directori
link	Obtindre informació del node-i d'un fitxer
unlink	Eliminar una entrada de directori

	Fitxers
open	Obrir/Crear fitxers
read	Lectura de fitxers
write	Escriptura de fitxers
close	Tancar fitxers
lseek	Posicionament de lectura/escriptura en fitxer
stat	Obtindre informació del node-i d'un fitxer
dup2	Duplica un descriptor de fitxer
pipe	Creació de tub
mkfifo	Creació de tub amb nom (fifo)

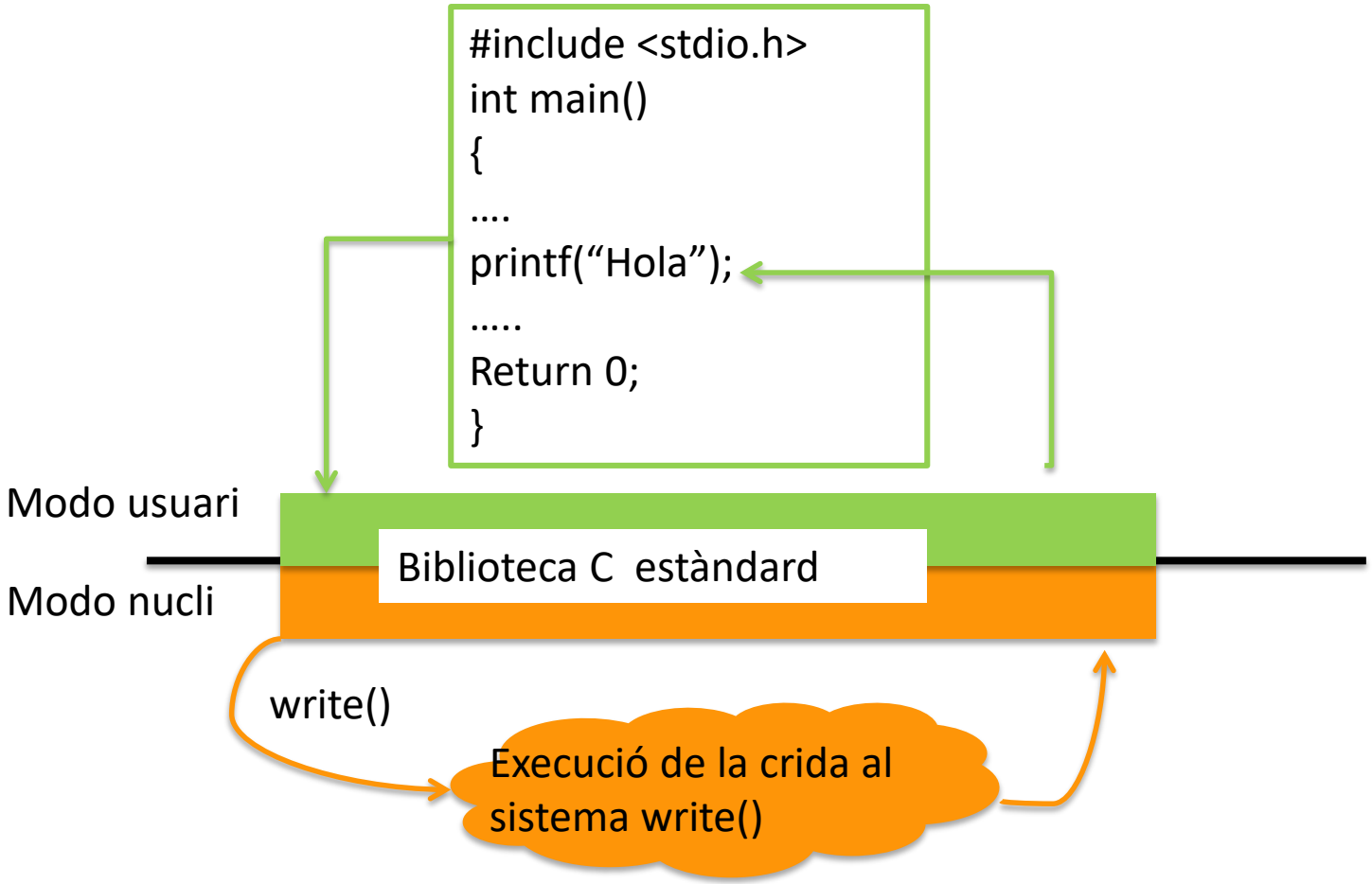
	Senyals
kill	Enviar senyals
alarm	Generar una alarma (senyal de rellotge)
sigemptyset	Iniciar una màscara per a que no tinga senyals seleccionades
sigfillset	Iniciar una máscara per a que continga tots els senyals
sigaddset	Afegir un senyal concret a un conjunt de senyals
sigdelset	Esborrar un senyal concret d'un conjunt de senyals
sigismember	Consultar si un senyal concret pertany a un conjunt de senyals
sigprocmask	Examinar /Modificar /Establir una màscara de senyals
sigaction	Capturar/Manejar un senyal
sigsuspend	Esperar la captura de senyals

Crides al sistema

- El **bit de mode** diferencia entre les tasques executades pel SO i les executades a nivell d'usuari
- Quan es fica en marxa el sistema, el hardware s'inicia en mode nucli (bit de mode =0), aleshores es carrega el sistema operatiu i s'inicien les aplicacions d'usuari en mode usuari (bit de mode=1)



- **Biblioteques del llenguatge C estàndard**
  - Proporcionen una interfície portable a moltes crides al sistema.



Contingut

- Arquitectura hardware del computador
- Interrupcions
- Modes d'execució
- Crides al sistema
- **Utilitats del sistema.**

- **Utilitats del sistema operatiu**
  - S'executen com a processos d'usuari i proporcionen **un entorn més còmode** de treball.
  - Es proporciona com a part del sistema operatiu, però no són imprescindibles per al funcionament de la màquina.
  - Exemples en UNIX
    - Tractament de fitxers: mkdir, cp, mv, ls ……
    - Filtres: grep, sort, head, tail ….
    - Editors, compiladors, assembladors, editors d'enllaços …
    - Sistemes de finestres: X11
    - Comunicacions: mail, ftp, rlogin ….
    - Intèrprets d'ordres: sh, ksh, bash

- Exemple "*my_copy*"

```
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
int main(int argc, char *argv[]){
  if (argc!=3) {
  fprintf(stderr,"Ús: %s f-origen"
    " f-destí\n", argv[0]);
  exit(1); }
  copy(argv[1],argv[2]);
  return 0;
}
```

```
#define BUFSIZE 1024

void copy(char *from, char *to)
{ int fromfd, tofd, nread;
  char buf[BUFSIZE];

  if ((fromfd = open(from, O_RDONLY)) == -1)
    { perror(from); exit(1) }
  if ((tofd = creat(to,0666)) == -1)
    { perror(to); exit(1); }
  while ((nread = read(fromfd, buf, sizeof (buf))) > 0)
      if (write(tofd, buf, nread) != nread)
        { perror("write"); exit(1); }
  if (nread == -1) perror("read");
  if (close(fromfd) == -1 || close(tofd) == -1)
      perror("close");
}
```