

# Unitat 1 · Sistemes Operatius (SO)

## Estructura dels Sistemes Operatius

---

Jordi Mateo [jordi.mateo@udl.cat](mailto:jordi.mateo@udl.cat)

Escola Politècnica Superior (EPS) <https://www.eps.udl.cat/> · Departament d'Enginyeria Informàtica i Disseny Digital <https://deidd.udl.cat/>

# Crides a sistema

2024-09-30

Unitat 1 · Sistemes Operatius (SO)  
└─Crides a sistema

Crides a sistema

# Que són les crides a sistema (System Calls)?

Les **crides al sistema** ofereixen les funcions bàsiques per a poder utilitzar i interactuar amb els recursos del sistema de manera correcta i controlada.

## Nº de crides a sistema

- Linux: aprox. 300
- FreeBSD: aprox. 500
- Windows: aprox. 2000

## Usos comuns

- Crear, obrir, tancar i eliminar fitxers.
- Crear i gestionar nous processos.
- Crear i gestionar la xarxa.

## Utilització

- Per accedir a les crides de sistema necessitem les **Llibreries del sistema**.
- Per exemple:
  - Unix: *stdio,stdlib,sys/shm,..*
  - Windows DLL: *system32.dll, ..*

Les **crides al sistema** ofereixen les funcions bàsiques per a poder utilitzar i interactuar amb els recursos del sistema de manera correcta i controlada.

### Nº de crides a sistema

- Linux: aprox. 300
- FreeBSD: aprox. 500
- Windows: aprox. 2000

### Usos comuns

- Crear, obrir, tancar i eliminar fitxers.
- Crear i gestionar nous processos.
- Crear i gestionar la xarxa.

### Utilització

- Per accedir a les crides de sistema necessitem les **Llibreries del sistema**.
- Per exemple:
  - Unix: *stdio,stdlib,sys/shm,..*
  - Windows DLL: *system32.dll, ..*

Quines són les principals crides a sistema?

	Windows	Unix
Control de processos	CreateProcess()	fork()
	ExitProcess()	exit()
	WaitForSingleObject()	wait()
Manteniment & Informació	GetCurrentProcessID()	getpid()
	SetTimer()	alarm()
	Sleep()	sleep()
Comunicació	CreatePipe()	pipe()
Protecció	SetFileSecurity()	chmod()
	SetSecurityDescriptorGroup()	chown()
Manipulació de fitxers	CreateFile()	open()
	ReadFile()	read()
	WriteFile()	write()
	CloseHandle()	close()

└─Crides a sistema

└─Quines són les principals crides a sistema?

2024-09-30

	Windows	Unix
Control de processos	CreateProcess()	fork()
	ExitProcess()	exit()
	WaitForSingleObject()	wait()
Manteniment & Informació	GetCurrentProcessID()	getpid()
	SetTimer()	alarm()
	Sleep()	sleep()
Comunicació	CreatePipe()	pipe()
Protecció	SetFileSecurity()	chmod()
	SetSecurityDescriptorGroup()	chown()
Manipulació de fitxers	CreateFile()	open()
	ReadFile()	read()
	WriteFile()	write()
	CloseHandle()	close()

Exemple amb la llibreria (stdio.h)

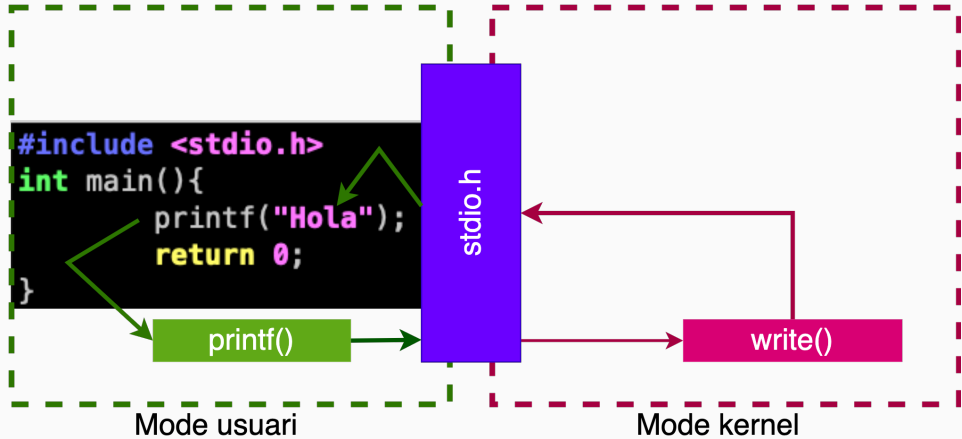
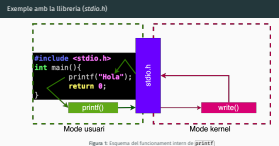


Figura 1: Esquema del funcionament intern de `printf`

2024-09-30

└ Crides a sistema

└ Exemple amb la llibreria (stdio.h)



# Nucli

---

2024-09-30

Unitat 1 · Sistemes Operatius (SO)  
└─ Nucli

Nucli

---

# Què és el nucli d'un sistema operatiu?

El nucli del **SO** és la capa més crítica i conté les rutines de gestió del sistema relacionades amb els recursos físic. Es troba sempre carregat a la Memòria.

## Funcionalitats del nucli

- El nucli té la capacitat d'assignar i desassignar la CPU als usuaris i processos per evitar que acaparin el recurs de forma ininterrompuda.

El nucli del **SO** és la capa més crítica i conté les rutines de gestió del sistema relacionades amb els recursos físic. Es troba sempre carregat a la Memòria.

Funcionalitats del nucli

- El nucli té la capacitat d'assignar i desassignar la CPU als usuaris i processos per evitar que acaparin el recurs de forma ininterrompuda.
- El nucli evita que els usuaris i processos accedeixin a dades d'altres usuaris.

- El nucli té la capacitat d'assignar i desassignar la CPU als usuaris i processos per evitar que acaparin el recurs de forma ininterrompuda.
- El nucli evita que els usuaris i processos accedeixin a dades d'altres usuaris.



El nucli del **SO** és la capa més crítica i conté les rutines de gestió del sistema relacionades amb els recursos físic. Es troba sempre carregat a la Memòria.

Funcionalitats del nucli

- El nucli té la capacitat d'assignar i desassignar la CPU als usuaris i processos per evitar que acaparin el recurs de forma ininterrompuda.
- El nucli evita que els usuaris i processos accedeixin a dades d'altres usuaris.
- El nucli evita els usuaris modifiquin el codi i les dades del nucli.

El nucli del **SO** és la capa més crítica i conté les rutines de gestió del sistema relacionades amb els recursos físic. Es troba sempre carregat a la Memòria.

Funcionalitats del nucli

- El nucli té la capacitat d'assignar i desassignar la CPU als usuaris i processos per evitar que acaparin el recurs de forma ininterrompuda.
- El nucli evita que els usuaris i processos accedeixin a dades d'altres usuaris.
- El nucli evita els usuaris modifiquin el codi i les dades del nucli.

El nucli del **SO** és la capa més crítica i conté les rutines de gestió del sistema relacionades amb els recursos físic. Es troba sempre carregat a la Memòria.

Funcionalitats del nucli

- El nucli té la capacitat d'assignar i desassignar la CPU als usuaris i processos per evitar que acaparin el recurs de forma ininterrompuda.
- El nucli evita que els usuaris i processos accedeixin a dades d'altres usuaris.
- El nucli evita els usuaris modifiquin el codi i les dades del nucli.
- El nucli evita que els usuaris realitzin E/S il·legals.

El nucli del **SO** és la capa més crítica i conté les rutines de gestió del sistema relacionades amb els recursos físic. Es troba sempre carregat a la Memòria.

Funcionalitats del nucli

- El nucli té la capacitat d'assignar i desassignar la CPU als usuaris i processos per evitar que acaparin el recurs de forma ininterrompuda.
- El nucli evita que els usuaris i processos accedeixin a dades d'altres usuaris.
- El nucli evita els usuaris modifiquin el codi i les dades del nucli.
- El nucli evita que els usuaris realitzin E/S il·legals.

Mode kernel

- El codi que s’executa en aquest mode té accés a qualsevol adreça de Memòria i a tots els recursos hardware.
- Si un programa falla en aquest mode, tot el sistema quedarà aturat.

Observació

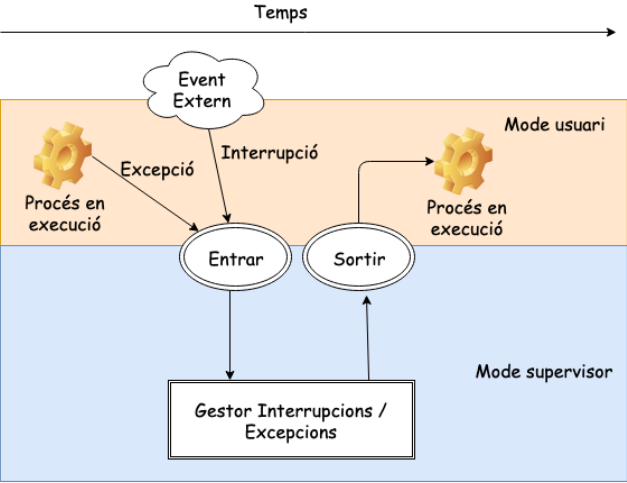
En mode usuari la CPU comprova cada instrucció per comprovar que el procés pot realitzar-la. En el mode kernel s’executen sense cap comprovació de protecció.

Mode usuari

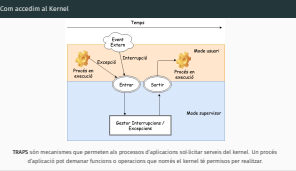
- El codi no té accés directe a Memòria ni als recursos hardware.
- Si un programa falla en aquest mode, únicament atura el programa i no el sistema.

Qué és la dualitat?	
<p><b>Mode kernel</b></p> <ul style="list-style-type: none"><li>• El codi que s'executa en aquest mode té accés a qualsevol adreça de Memòria i a tots els recursos hardware.</li><li>• Si un programa falla en aquest mode, tot el sistema quedarà aturat.</li></ul>	<p><b>Mode usuari</b></p> <ul style="list-style-type: none"><li>• El codi no té accés directe a Memòria ni als recursos hardware.</li><li>• Si un programa falla en aquest mode, únicament atura el programa i no el sistema.</li></ul>
<p><b>Observació</b></p> <p>En mode usuari la CPU comprova cada instrucció per comprovar que el procés pot realitzar-la. En el mode kernel s'executen sense cap comprovació de protecció.</p>	

# Com accedim al Kernel



TRAPS són mecanismes que permeten als processos d'aplicacions sol·licitar serveis del kernel. Un procés d'aplicació pot demanar funcions o operacions que només el kernel té permisos per realitzar.



Exemple de trap: `read(fd, buffer, nbytes)`

Espai d'usuari

- 1. Programa C:
  - Posa *fd, buffer, nbytes* als registres(*RDI,RSI,RDX*).
  - Crida a `read()`.
- 2. Llibreria de sistema:
  - Posa el codi de `read` a *RAX*.
  - Executa `syscall` (TRAP).

Transició a mode kernel

Espai del kernel

- Crida al planificador de tasques.
- Localitza el handler de `read`.
- Executa el handler de `read`:
  - Llegeix dades del dispositiu d'E/S.
  - Col·loca les dades al buffer d'usuari.
- Retorna a l'espai d'usuari.

Transició a mode usuari

Espai d'usuari

- **Llibreria de sistema:**  
Retorna el resultat de la crida a `read()` al Programa C.
- **Programa:** Continua l'execució utilitzant les dades llegides. O bé, tracta l'error si n'hi ha (*errno*)

2024-09-30

└─Nucli

└─Exemple de trap: `read(fd, buffer, nbytes)`



# Què és una interrupció?

Les interrupcions són esdeveniments **hardware o software** asíncrons independents al procés que actualment s'està executant.

## Gestió

- 1. Guardar el context del procés actual.
- 2. Canviar a mode kernel.
- 3. Determinar la causa de la interrupció.
- 4. Buscar la direcció de la RTI.
- 5. Cridar a la RTI.
- 6. Restaurar el context del procés.
- 7. Tornar a mode usuari

El seu tractament és **prioritari**.

2024-09-30

Unitat 1 · Sistemes Operatius (SO)

└─ Nucli

└─ Què és una interrupció?

Què és una interrupció?

Les interrupcions són esdeveniments **hardware o software** asíncrons independents al procés que actualment s'està executant.

Gestió

1. Guardar el context del procés actual.

2. Canviar a mode kernel.

3. Determinar la causa de la interrupció.

4. Buscar la direcció de la RTI.

5. Cridar a la RTI.

6. Restaurar el context del procés.

7. Tornar a mode usuari

El seu tractament és **prioritari**.

Generalment són errors. Trencaments de la seqüència no previstos provocats per l'execució en curs de l'usuari (condició anormal mentre s'executa una instrucció.). El Sistema operatiu intenta tractar l'excepció si no pot solucionar el problema, s'enviarà un senyal al procés. El procés tracta el senyal executant el gestor per defecte o específic del procés per aquell senyal.

Faults

Un fault és un tipus d'excepció que s'informa abans de l'execució de la instrucció i que normalment es pot corregir. (per exemple, error de pàgina).

Traps

Una trap s'informa després de l'execució de la instrucció en què s'ha detectat l'excepció. (per exemple, trap de depuració).

Aborts

Errors que no es poden corregir.

2024-09-30

Excepcions		
Generalment són errors. Trencaments de la seqüència no previstos provocats per l'execució en curs de l'usuari (condició anormal mentre s'executa una instrucció.). El Sistema operatiu intenta tractar l'excepció si no pot solucionar el problema, s'enviarà un senyal al procés. El procés tracta el senyal executant el gestor per defecte o específic del procés per aquell senyal.		
<b>Faults</b> Un fault és un tipus d'excepció que s'informa abans de l'execució de la instrucció i que normalment es pot corregir. (per exemple, error de pàgina).	<b>Traps</b> Una trap s'informa després de l'execució de la instrucció en què s'ha detectat l'excepció. (per exemple, trap de depuració).	<b>Aborts</b> Errors que no es poden corregir.

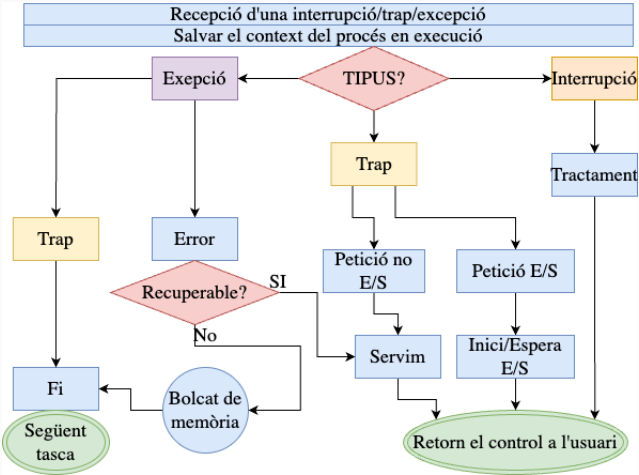
Name	Type
Divide-by-zero	Fault
Debug	Fault/Trap
Non-maskable Interrupt	Interrupt
Breakpoint	Trap
Overflow	Trap
Bound Range Exceeded	Fault
Device not Available	Fault
Double Fault	Abort
Segment Not Present	Fault
Page Fault	Fault

Per a més informació podeu consultar el següent enllaç: [Excepcions](#).

Name	Type
Divide-by-zero	Fault
Debug	Fault/Trap
Non-maskable Interrupt	Interrupt
Breakpoint	Trap
Overflow	Trap
Bound Range Exceeded	Fault
Device not Available	Fault
Double Fault	Abort
Segment Not Present	Fault
Page Fault	Fault

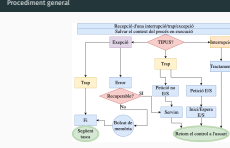


Procediment general

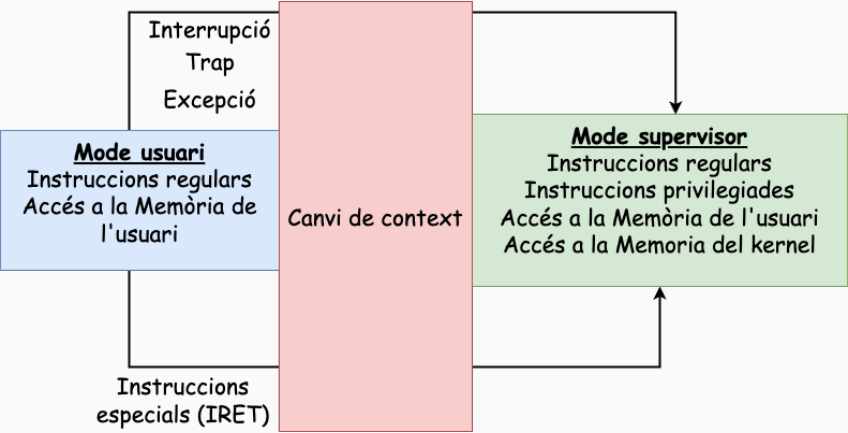


2024-09-30

Unitat 1 · Sistemes Operatius (SO)  
└─ Nucli  
└─ Procediment general



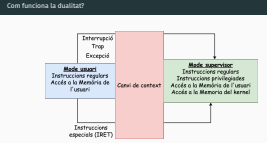
# Com funciona la dualitat?



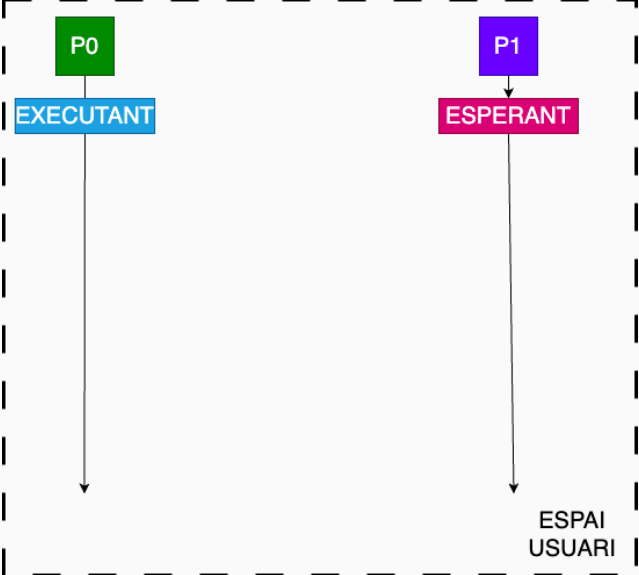
2024-09-30

Nucli

Com funciona la dualitat?

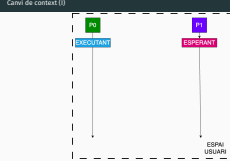


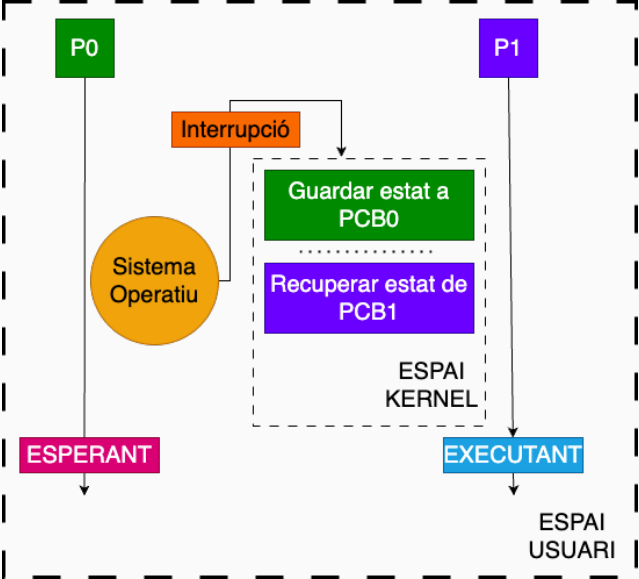
Canvi de context (I)



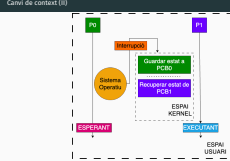
2024-09-30

Unitat 1 · Sistemes Operatius (SO)  
└─ Nucli  
└─ Canvi de context (I)





2024-09-30



# Quines són les característiques de la dualitat

- El codi que pertany al nucli sempre s’executa amb tots els privilegis (**mode supervisor**).

Unitat 1 · Sistemes Operatius (SO)

└─ Nucli

└─ Quines són les característiques de la dualitat

Quines són les característiques de la dualitat

- El codi que pertany al nucli sempre s’executa amb tots els privilegis (**mode supervisor**).

# Quines són les característiques de la dualitat

- El codi que pertany al nucli sempre s’executa amb tots els privilegis (**mode supervisor**).
- El SO minimitza el temps i el codi que s’executa aquest mode.

Unitat 1 · Sistemes Operatius (SO)

└─ Nucli

└─ Quines són les característiques de la dualitat

Quines són les característiques de la dualitat

- El codi que pertany al nucli sempre s’executa amb tots els privilegis (**mode supervisor**).
- El SO minimitza el temps i el codi que s’executa aquest mode.

# Quines són les característiques de la dualitat

- El codi que pertany al nucli sempre s’executa amb tots els privilegis (**mode supervisor**).
- El SO minimitza el temps i el codi que s’executa aquest mode.
- Per accedir al nucli necessitem realitzar un **canvi de context**.

- El codi que pertany al nucli sempre s’executa amb tots els privilegis (**mode supervisor**).
- El SO minimitza el temps i el codi que s’executa aquest mode.
- Per accedir al nucli necessitem realitzar un **canvi de context**.

# Quines són les característiques de la dualitat

- El codi que pertany al nucli sempre s’executa amb tots els privilegis (**mode supervisor**).
- El SO minimitza el temps i el codi que s’executa aquest mode.
- Per accedir al nucli necessitem realitzar un **canvi de context**.
- Les **crides a sistema** permeten utilitzar les rutines del nucli.

- El codi que pertany al nucli sempre s’executa amb tots els privilegis (**mode supervisor**).
- El SO minimitza el temps i el codi que s’executa aquest mode.
- Per accedir al nucli necessitem realitzar un **canvi de context**.
- Les **crides a sistema** permeten utilitzar les rutines del nucli.



# Quines són les característiques de la dualitat

- El codi que pertany al nucli sempre s’executa amb tots els privilegis (**mode supervisor**).
- El SO minimitza el temps i el codi que s’executa aquest mode.
- Per accedir al nucli necessitem realitzar un **canvi de context**.
- Les **crides a sistema** permeten utilitzar les rutines del nucli.
- **Instruccions privilegiades**. Totes les instruccions potencialment insegures estan prohibides quan s’executen en mode d’usuari.

# Quines són les característiques de la dualitat

- El codi que pertany al nucli sempre s’executa amb tots els privilegis (**mode supervisor**).
- El SO minimitza el temps i el codi que s’executa aquest mode.
- Per accedir al nucli necessitem realitzar un **canvi de context**.
- Les **crides a sistema** permeten utilitzar les rutines del nucli.
- **Instruccions privilegiades**. Totes les instruccions potencialment insegures estan prohibides quan s’executen en mode d’usuari.
- **Protecció de memòria**. Tots els accessos a la memòria fora de la regió de memòria vàlida d’un procés estan prohibits quan s’executen en mode d’usuari. Transforma les referències lògiques (abstractes) en físiques (concretes).

- El codi que pertany al nucli sempre s’executa amb tots els privilegis (**mode supervisor**).
- El SO minimitza el temps i el codi que s’executa aquest mode.
- Per accedir al nucli necessitem realitzar un **canvi de context**.
- Les **crides a sistema** permeten utilitzar les rutines del nucli.
- **Instruccions privilegiades**. Totes les instruccions potencialment insegures estan prohibides quan s’executen en mode d’usuari.
- **Protecció de memòria**. Tots els accessos a la memòria fora de la regió de memòria vàlida d’un procés estan prohibits quan s’executen en mode d’usuari. Transforma les referències lògiques (abstractes) en físiques (concretes).

# Quines són les característiques de la dualitat

- El codi que pertany al nucli sempre s’executa amb tots els privilegis (**mode supervisor**).
- El SO minimitza el temps i el codi que s’executa aquest mode.
- Per accedir al nucli necessitem realitzar un **canvi de context**.
- Les **crides a sistema** permeten utilitzar les rutines del nucli.
- **Instruccions privilegiades**. Totes les instruccions potencialment insegures estan prohibides quan s’executen en mode d’usuari.
- **Protecció de memòria**. Tots els accessos a la memòria fora de la regió de memòria vàlida d’un procés estan prohibits quan s’executen en mode d’usuari. Transforma les referències lògiques (abstractes) en físiques (concretes).
- **Interrupció del temporitzador**. Independentment del que faci el procés. El nucli pot recuperar periòdicament el control del procés actual. Errors que no es poden corregir.

- El codi que pertany al nucli sempre s’executa amb tots els privilegis (**mode supervisor**).
- El SO minimitza el temps i el codi que s’executa aquest mode.
- Per accedir al nucli necessitem realitzar un **canvi de context**.
- Les **crides a sistema** permeten utilitzar les rutines del nucli.
- **Instruccions privilegiades**. Totes les instruccions potencialment insegures estan prohibides quan s’executen en mode d’usuari.
- **Protecció de memòria**. Tots els accessos a la memòria fora de la regió de memòria vàlida d’un procés estan prohibits quan s’executen en mode d’usuari. Transforma les referències lògiques (abstractes) en físiques (concretes).
- **Interrupció del temporitzador**. Independentment del que faci el procés. El nucli pot recuperar periòdicament el control del procés actual. Errors que no es poden corregir.

- Assignació d'adreces de memòria.

2024-09-30

Unitat 1 · Sistemes Operatius (SO)

└─ Nucli

└─ Exemples d'instruccions privilegiades

- Assignació d'adreces de memòria.

- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.

2024-09-30

Unitat 1 · Sistemes Operatius (SO)

└─ Nucli

└─ Exemples d'instruccions privilegiades

- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.

- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.
- Invalideu les entrades a les taules de pàgines.

2024-09-30

- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.
- Invalideu les entrades a les taules de pàgines.

- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.
- Invalideu les entrades a les taules de pàgines.
- Carregueu i llegiu els registres del sistema.

- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.
- Invalideu les entrades a les taules de pàgines.
- Carregueu i llegiu els registres del sistema.

- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.
- Invalideu les entrades a les taules de pàgines.
- Carregueu i llegiu els registres del sistema.
- Canvieu els modes de processador del nucli a l'usuari.



- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.
- Invalideu les entrades a les taules de pàgines.
- Carregueu i llegiu els registres del sistema.
- Canvieu els modes de processador del nucli a l'usuari.
- Canvieu el voltatge i la freqüència del processador.

- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.
- Invalideu les entrades a les taules de pàgines.
- Carregueu i llegiu els registres del sistema.
- Canvieu els modes de processador del nucli a l'usuari.
- Canvieu el voltatge i la freqüència del processador.
- Parar un processador.

- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.
- Invalideu les entrades a les taules de pàgines.
- Carregueu i llegiu els registres del sistema.
- Canvieu els modes de processador del nucli a l'usuari.
- Canvieu el voltatge i la freqüència del processador.
- Parar un processador.

- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.
- Invalideu les entrades a les taules de pàgines.
- Carregueu i llegiu els registres del sistema.
- Canvieu els modes de processador del nucli a l'usuari.
- Canvieu el voltatge i la freqüència del processador.
- Parar un processador.
- Restableix un processador.

- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.
- Invalideu les entrades a les taules de pàgines.
- Carregueu i llegiu els registres del sistema.
- Canvieu els modes de processador del nucli a l'usuari.
- Canvieu el voltatge i la freqüència del processador.
- Parar un processador.
- Restableix un processador.

- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.
- Invalideu les entrades a les taules de pàgines.
- Carregueu i llegiu els registres del sistema.
- Canvieu els modes de processador del nucli a l'usuari.
- Canvieu el voltatge i la freqüència del processador.
- Parar un processador.
- Restableix un processador.
- Fer operacions d'E/S.

- Assignació d'adreces de memòria.
- Esborreu o invalideu la memòria cau de dades.
- Invalideu les entrades a les taules de pàgines.
- Carregueu i llegiu els registres del sistema.
- Canvieu els modes de processador del nucli a l'usuari.
- Canvieu el voltatge i la freqüència del processador.
- Parar un processador.
- Restableix un processador.
- Fer operacions d'E/S.

2024-09-30

# Estructures dels sistemes operatius

---

Quina és la millor manera d’organitzar/separar totes les parts del sistema operatiu?

Reptes

- Com **organitzem** les parts?
- **Definició de polítiques:** Quines seran les accions a realitzar.
- Com **cooperen** les parts?
- **Mecanismes:** Com es duran a terme aquestes accions.

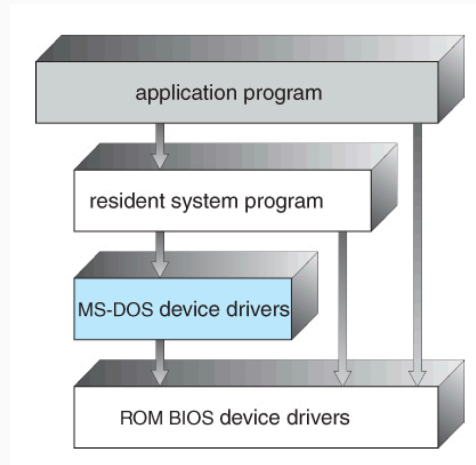
Classificació (Estructura Interna)

1. Monolítics
2. Capes
3. Micro-kernel
4. Híbrids
5. Màquines Virtuals

Quina és la problematica?	
Quina és la millor manera d'organitzar/separar totes les parts del sistema operatiu?	
Reptes	Classificació (Estructura Interna)
<ul style="list-style-type: none"><li>• Com <b>organitzem</b> les parts?</li><li>• <b>Definició de polítiques:</b> Quines seran les accions a realitzar.</li><li>• Com <b>cooperen</b> les parts?</li><li>• <b>Mecanismes:</b> Com es duran a terme aquestes accions.</li></ul>	<ol style="list-style-type: none"><li>1. Monolítics</li><li>2. Capes</li><li>3. Micro-kernel</li><li>4. Híbrids</li><li>5. Màquines Virtuals</li></ol>

## Característiques

- Estructura no ben definida.
- El **SO** és un conjunt de procediments que es poden cridar sense cap limitació.
- Son sistemes complexos; difícil d'implementar i de depurar.
- No tenen mode **dual**.
- Exemple: MS-DOS

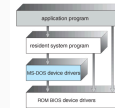


Els sistemes operatius com MS-DOS o Unix (original) no tenien estructures ben definides.

No hi havia cap mode d'execució de la CPU (usuari i nucli), de manera que els errors en les aplicacions podrien provocar un bloqueig de tot el sistema.

Quan es va escriure DOS originalment, els seus desenvolupadors no tenien ni idea de quant de gran i important esdevindria. Va ser escrit per uns quants programadors en un temps relativament curt, sense el benefici de les tècniques modernes d'enginyeria de programari, i després va anar creixent amb el pas del temps fins a superar les seves expectatives originals. No divideix el sistema en subsistemes i no distingeix entre modes d'usuari i nucli, cosa que permet a tots els programes accedir directament al maquinari subjacent.

- Característiques
- Estructura no ben definida.
  - El **SO** és un conjunt de procediments que es poden cridar sense cap limitació.
  - Son sistemes complexos; difícil d'implementar i de depurar.
  - No tenen mode **dual**.
  - Exemple: MS-DOS



Característiques de l'estructura monolítica

Els serveis d'usuari i serveis del kernel s'implementen sota el mateix espai d'adreces.

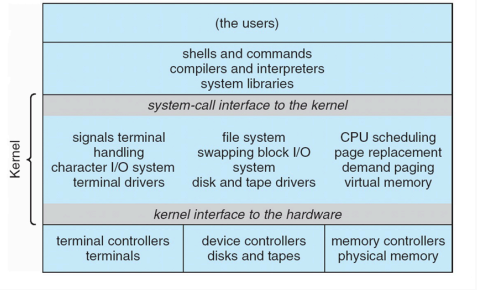


Figura 2: Esquema d'un kernel monolític

Pros de l'estructura monolítica

- Les funcionalitats (serveis) s'invoquen amb crides al sistema.
- Els controladors de dispositius es carreguen al nucli i passen a formar part del nucli.

Cons de l'estructura monolítica

- Difícil d'entendre, modificar i mantenir.
- Poc fiable (sense aïllament entre els mòduls del sistema)

Estructura Monolítica

Característiques de l'estructura monolítica

Els serveis d'usuari i serveis del kernel s'implementen sota el mateix espai d'adreces.

Diagrama d'un kernel monolític:

- (the users)**
- shells and commands, compilers and interpreters, system libraries**
- system-call interface to the kernel**
- Kernel** (bracketed section):
  - signals terminal handling, character I/O system, terminal drivers
  - file system, swapping block I/O system, disk and tape drivers
  - CPU scheduling, page replacement, demand paging, virtual memory
- kernel interface to the hardware**
- terminal controllers terminals, device controllers disks and tapes, memory controllers physical memory**

Figura 2: Esquema d'un kernel monolític

Pros de l'estructura monolítica

- Les funcionalitats (serveis) s'invoquen amb crides al sistema.
- Els controladors de dispositius es carreguen al nucli i passen a formar part del nucli.

Cons de l'estructura monolítica

- Difícil d'entendre, modificar i mantenir.
- Poc fiable (sense aïllament entre els mòduls del sistema)



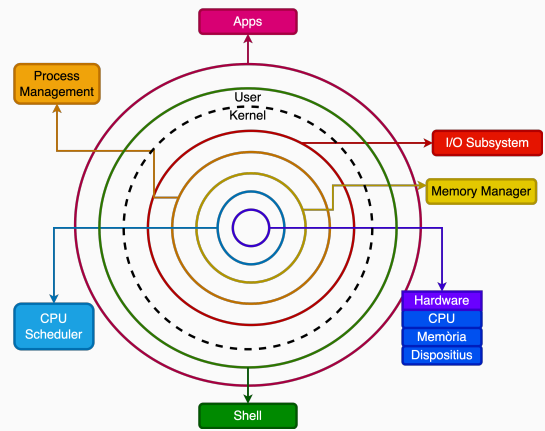


Figura 3: Esquema del kernel per capes

Pros de l'estructura per capes

- Independència entre les capes.
- Permet descriure el **SO** de forma clara.
- Simplicitat en la construcció i depuració.

Cons de l'estructura per capes

- Rendiment.
- És difícil definir les capes a causa de les limitacions per comunicar-se.

2024-09-30

Un exemple és UNIX, aquest sistema operatiu, creat per Dennis Ritchie i Ken Thompson als anys 70, va adoptar una arquitectura en capes més simple que Multics. Aquestes capes són: Hardware, Kernel, Shell i Aplicacions. Un exemple és: NetBSD.

Permet que cada anell tingui un conjunt de funcions i responsabilitats clarament definides, i que cada anell pugui comunicar-se amb els anells adjacents.

Estructura per capes

Pros de l'estructura per capes

- Independència entre les capes.
- Permet descriure el SO de forma clara.
- Simplicitat en la construcció i depuració.

Cons de l'estructura per capes

- Rendiment.
- És difícil definir les capes a causa de les limitacions per comunicar-se.

Figura 3: Esquema del kernel per capes

## Característiques de l'estructura Microkernel

- Els serveis d'usuari i serveis del kernel s'implementi en diferents espais d'adreces.
- Comunicació entre els mòduls utilitza el pas de missatges.

## Pros de l'estructura Microkernel

- El kernel té una mida més reduïda.
- Portable, segur, fiable i extensible.

## Cons de l'estructura Microkernel

- Reducció de la velocitat d'execució i del rendiment.

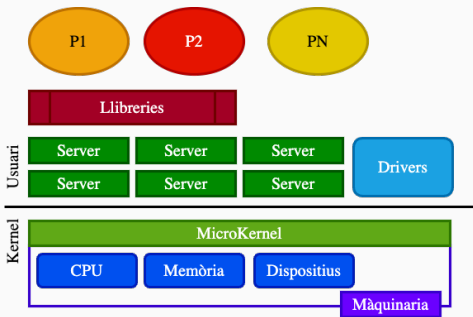


Figura 4: Esquema d'un microkernel

2024-09-30

El sistema operatiu MacOS, desenvolupat per Apple, va aprofitar l'estabilitat i seguretat del nucli Mach com a base per al seu sistema operatiu. En aquest sistema, serveis com la gestió de memòria, la gestió de fitxers i la xarxa es van traslladar fora del nucli, a servidors externs, millorant la estabilitat, fiabilitat i modularitat del sistema. Sistemes en temps real com QNX, o dispositius encastrats poden utilitzar aquesta arquitectura.

Estructura Microkernel

Característiques de l'estructura Microkernel

- Els serveis d'usuari i serveis del kernel s'implementi en diferents espais d'adreces.
- Comunicació entre els mòduls utilitza el pas de missatges.

Pros de l'estructura Microkernel

- El kernel té una mida més reduïda.
- Portable, segur, fiable i extensible.

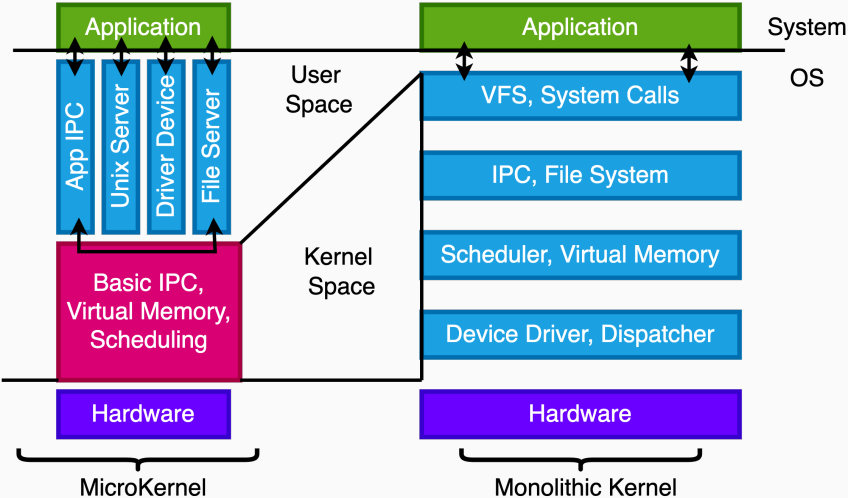
Cons de l'estructura Microkernel

- Reducció de la velocitat d'execució i del rendiment.

Diagrama d'estructura Microkernel. A la part superior, tres processos d'usuari (P1, P2, PN) són representats per ovals groc, vermell i groc respectivament. Sota d'ells, una barreja de color vermell conté el text 'Lliberies'. A continuació, una secció etiquetada 'Usuari' conté una matrija de components: tres columnes de 'Server' (verds) i una columna de 'Drivers' (blau). A sota d'aquesta, una línia horitzontal separa la part 'Usuari' de la part 'Kernel'. Dins de la secció 'Kernel', hi ha una barreja de color verd etiquetada 'MicroKernel'. Sota d'aquesta, hi ha tres components: 'CPU' (blau), 'Memòria' (blau) i 'Dispositius' (blau). A la part inferior dreta, una barreja de color morat etiquetada 'Màquinaria' està connectada a la secció 'Kernel'.

Figura 4: Esquema d'un microkernel

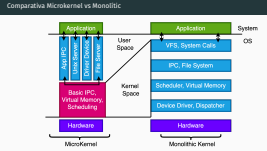
# Comparativa Microkernel vs Monolític



2024-09-30

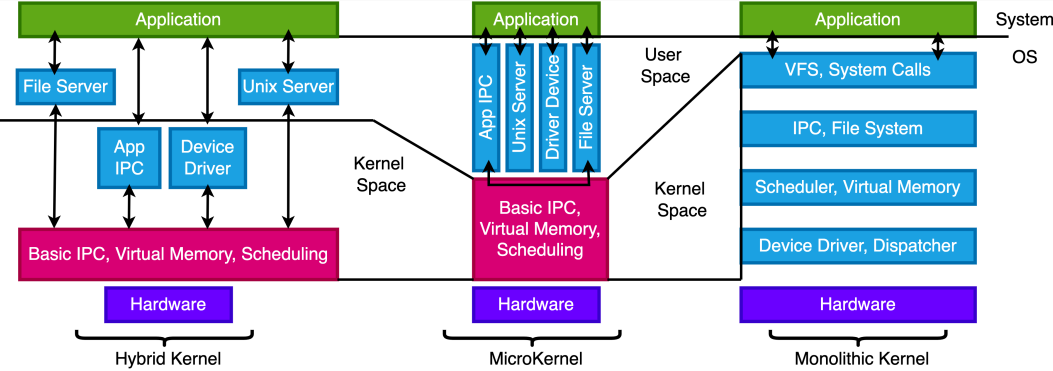
Unitat 1 · Sistemes Operatius (SO)  
└ Estuctures dels sistemes operatius

└ Comparativa Microkernel vs Monolític



El **microkernel** és més *lent* però més *segur i fiable* que el nucli **monolític**. El nucli **monolític** és *ràpid* però menys *segur*, ja que qualsevol fallada del servei pot causar un bloqueig del sistema.

# Comparativa Microkernel vs Monolític vs Híbrids

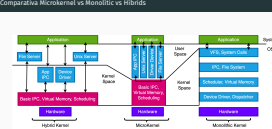


2024-09-30

Unitat 1 · Sistemes Operatius (SO)

└─ Estructures dels sistemes operatius

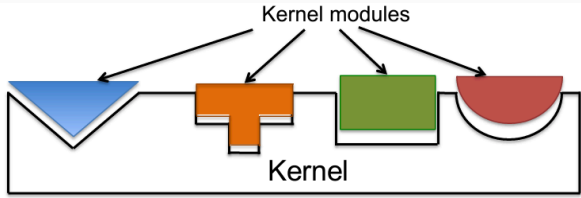
└─ Comparativa Microkernel vs Monolític vs Híbrids



## Definició

El serveis (*core*) estan integrats al kernel, la resta es poden carregar i descarregar de forma dinàmica.

- No cal reiniciar per afegir nous mòduls.
- No cal implementar mecanismes de pas de missatges com en els microkernels.
- Qualsevol mòdul pot comunicar-se amb qualsevol altre.



2024-09-30

**Definició**  
El serveis (*core*) estan integrats al kernel, la resta es poden carregar i descarregar de forma dinàmica.

- No cal reiniciar per afegir nous mòduls.
- No cal implementar mecanismes de pas de missatges com en els microkernels.
- Qualsevol mòdul pot comunicar-se amb qualsevol altre.

Els sistemes operatius modulars com la majoria de sistemes operatius monolítics moderns com Linux, BSD, poden carregar (i descarregar) dinàmicament mòduls executables en temps d'execució.

Aquesta modularitat del sistema operatiu és a nivell binari (imatge) i no a nivell d'arquitectura.

Pràcticament, carregar mòduls dinàmicament és simplement una manera més flexible de manejar la imatge del sistema operatiu en temps d'execució, en lloc de reiniciar-lo amb una imatge diferent del sistema operatiu.

Els mòduls permeten ampliar fàcilment les capacitats dels sistemes operatius segons sigui necessari.

Els mòduls que es poden carregar dinàmicament comporten una petita sobrecàrrega en comparació amb la incorporació del mòdul a la imatge del sistema operatiu.

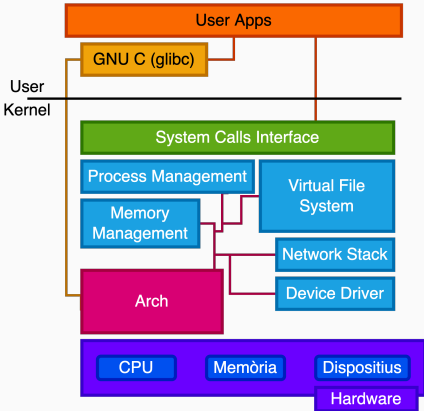
Tanmateix, en alguns casos, carregar mòduls dinàmicament (segons calgui) ajuda a mantenir la quantitat de codi que s'executa a l'espai del nucli al mínim; per exemple, per minimitzar la petjada del sistema operatiu per a dispositius incrustats o aquells amb recursos de maquinari limitats. És a dir, no cal que un mòdul descarregat s'emmagatzemi en memòria d'accés aleatori escàs.

# Exemple: Linux

## Definició

El nucli Linux és un dels projectes de codi obert més grans del món, amb milers de desenvolupadors que aporten codi i milions de línies de codi canviats per a cada versió.

- Arquitectura **Monolítica** híbrida basada en mòduls.
  - Enllaçat dinàmic.
  - Mòduls apilables.
- Disseny orientat a objectes.
- Suport per a múltiples fils d'execució.
- Suport per processament múltiple simètric.
- Abstracció hardware.



## Unitat 1 · Sistemes Operatius (SO)

### ↳ Estructures dels sistemes operatius

### ↳ Exemple: Linux

Es distribueix sota la llicència GPLv2, que simplement esmenta, requereix que qualsevol modificació del nucli feta amb el programari que s'envia al client s'hagi de posar a la seva disposició (els clients), tot i que a la pràctica la majoria de les empreses posen el codi font a disposició del públic.

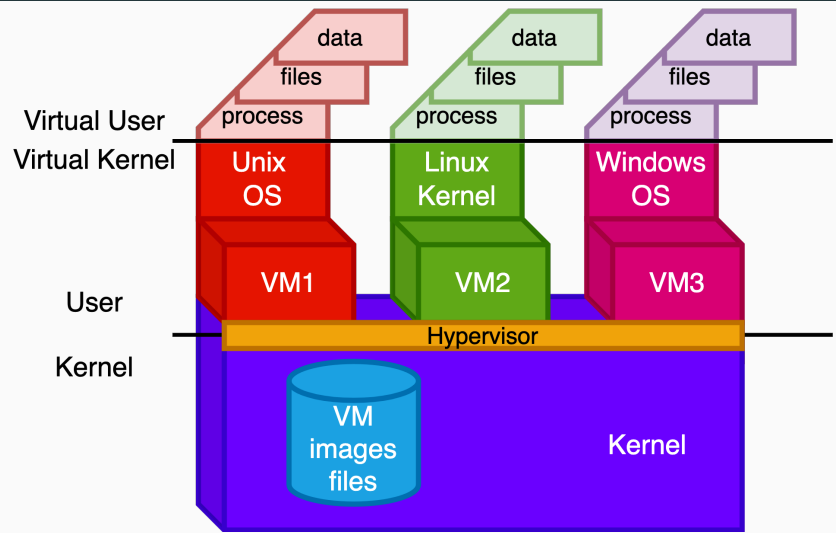
Per tal d'escalar el procés de desenvolupament, Linux utilitza un model de manteniment jeràrquic:

Linus Torvalds és el mantenidor del nucli Linux i fa merge dels **pull requests** de la comunitat.

La comunitat té un o més mantenidors que accepten *patches* de desenvolupadors, mantenint el seu propi arbre git.

- Linux Torvalds: git
- David Miller (treball en xarxa): git

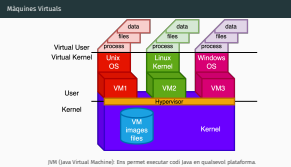




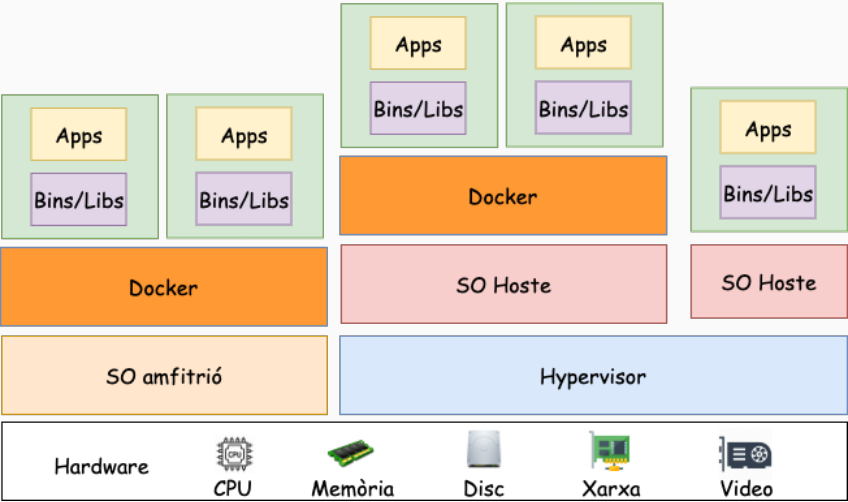
JVM (Java Virtual Machine): Ens permet executar codi Java en qualsevol plataforma.

2024-09-30

Unitat 1 · Sistemes Operatius (SO)  
└─ Estructures dels sistemes operatius  
  
└─ Màquines Virtuals

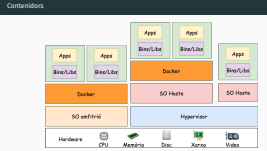


JVM (Java Virtual Machine): Ens permet executar codi Java en qualsevol plataforma.



2024-09-30

Unitat 1 · Sistemes Operatius (SO)  
└─ Estructures dels sistemes operatius  
  
└─ Contenidors





Arquitectura **minimalista** que intenta proporcionar als programes un control directe sobre el maquinari. A diferència dels microkernels o dels kernels monolítics, els exokernels no intenten abstraure gairebé res del maquinari, sinó que ofereixen primitives de baix nivell per accedir directament als recursos de la màquina. La gestió de recursos, com la memòria i la CPU, queda delegada als programes d'usuari o a les seves llibreries.

Imaginem que particionem una màquina en diferents màquines virtuals amb un subconjunt de recursos. Cada màquina virtual té un exokernel que li permet accedir directament als recursos de la màquina física. La idea és reduir el overhead del sistema separant la multiprogramació de les funcions del sistema operatiu en l'espai usuari.

En aquest cas no estem movent instruccions de mode kernel a mode usuari, sinó que estem movent la gestió de recursos del sistema operatiu a l'espai d'usuari. De fet, els exokernels són tan petits que poden ser incrustats en aplicacions específiques. Per exemple, un exokernel podria ser incrustat en una aplicació de control de trànsit per coordinar els semàfors d'una ciutat.

Exokernels

Arquitectura **minimalista** que intenta proporcionar als programes un control directe sobre el maquinari. A diferència dels microkernels o dels kernels monolítics, els exokernels no intenten abstraure gairebé res del maquinari, sinó que ofereixen primitives de baix nivell per accedir directament als recursos de la màquina. La gestió de recursos, com la memòria i la CPU, queda delegada als programes d'usuari o a les seves llibreries.

Imaginem que particionem una màquina en diferents màquines virtuals amb un subconjunt de recursos. Cada màquina virtual té un exokernel que li permet accedir directament als recursos de la màquina física. La idea és reduir el overhead del sistema separant la multiprogramació de les funcions del sistema operatiu en l'espai usuari.

Els unikernels són una forma de sistema operatiu que empaqueta tot el codi necessari per a una aplicació en un sol paquet. Aquest paquet s’executa directament sobre una màquina virtual o un hipervisor, sense cap sistema operatiu subministrat. Això permet als unikernels ser molt petits i molt eficients, ja que no hi ha cap codi innecessari.

S’inspiren en els contenidors, però en lloc d’executar-se en un sistema operatiu complet, s’executen directament sobre el hardware. Això els fa molt més petits i molt més eficients que els contenidors, ja que no hi ha cap sistema operatiu subministrat.

S’inspiren en els exokernels, però en lloc de proporcionar un conjunt de primitives de baix nivell per accedir directament als recursos de la màquina, proporcionen un conjunt de primitives de baix nivell per accedir directament als recursos de l’aplicació.

PREGUNTES?

Materials del curs

- Organització — OS-GEI-IGUALADA-2425
- Materials — Materials del curs
- Laboratoris — Laboratoris
- Recursos — Campus Virtual

**TAKE HOME MESSAGE:** La dualitat i el disseny d'estructures eficients són crucials per a una gestió segura i òptima dels recursos i processos, garantint alhora un funcionament estable i eficaç dels dispositius informàtics.



Figura 5: Això és tot per avui

2024-09-30

Això és tot per avui

PREGUNTES?

Materials del curs

- Organització — OS-GEI-IGUALADA-2425
- Materials — Materials del curs
- Laboratoris — Laboratoris
- Recursos — Campus Virtual

**TAKE HOME MESSAGE:** La dualitat i el disseny d'estructures eficients són crucials per a una gestió segura i òptima dels recursos i processos, garantint alhora un funcionament estable i eficaç dels dispositius informàtics.




Figura 5: Això és tot per avui