

Unitat 0 · Sistemes Operatius (SO)

Presentació del curs & Introducció a la temàtica

Jordi Mateo jordi.mateo@udl.cat

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

Currículum

- Enginyer i Doctor en Informàtica per la UdL.

2024-09-30

└─ Presentacions - *Qui soc?*

- Enginyer i Doctor en Informàtica per la UdL.

Currículum

- Enginyer i Doctor en Informàtica per la UdL.

Docència

- Professor **lector** a la UdL des de 2019.
- Sistemes Operatius (*GTIDIC, GEI*).
- Administració de Sistemes (*GTIDIC, GEI*).
- Desenvolupament d'Aplicacions per a dispositius mòvils (*GTIDIC*).
- Cloud Compunting (*Master in Health Data Science*)
- High Performance Computing (*Màster en Enginyeria Informàtica*)

Presentacions - Qui soc?
<div>Currículum</div> <div><ul style="list-style-type: none">• Enginyer i Doctor en Informàtica per la UdL.</div>
<div>Docència</div> <div><ul style="list-style-type: none">• Professor lector a la UdL des de 2019.• Sistemes Operatius (<i>GTIDIC, GEI</i>).• Administració de Sistemes (<i>GTIDIC, GEI</i>).• Desenvolupament d'Aplicacions per a dispositius mòvils (<i>GTIDIC</i>).• Cloud Computing (<i>Master in Health Data Science</i>)• High Performance Computing (<i>Màster en Enginyeria Informàtica</i>)</div>

Currículum

- Enginyer i Doctor en Informàtica per la UdL.

Docència

- Professor **lector** a la UdL des de 2019.
- Sistemes Operatius (*GTIDIC, GEI*).
- Administració de Sistemes (*GTIDIC, GEI*).
- Desenvolupament d'Aplicacions per a dispositus mòbils (*GTIDIC*).
- Cloud Compunting (*Master in Health Data Science*)
- High Performance Computing (*Màster en Enginyeria Informàtica*)

Recerca

- Membre del grup de recerca **Grup de computació distribuïda** des de 2012 (Universitat de Lleida).
- Membre del grup de recerca **Essence: Data Engineering & Distributed Computing Systems** des de 2022 (Universitat de Glasgow).
- Computació distribuïda: *Cloud, Edge i Fog Computing*.
- Camps d'aplicació: *Salut electrònica i Energia*.

2024-09-30

Presentacions - Qui soc?	
<div><p>Currículum</p><ul style="list-style-type: none">• Enginyer i Doctor en Informàtica per la UdL.<p>Docència</p><ul style="list-style-type: none">• Professor lector a la UdL des de 2019.• Sistemes Operatius (<i>GTIDIC, GEI</i>).• Administració de Sistemes (<i>GTIDIC, GEI</i>).• Desenvolupament d'Aplicacions per a dispositus mòbils (<i>GTIDIC</i>).• Cloud Computing (<i>Master in Health Data Science</i>)• High Performance Computing (<i>Màster en Enginyeria Informàtica</i>)</div>	<div><p>Recerca</p><ul style="list-style-type: none">• Membre del grup de recerca Grup de computació distribuïda des de 2012 (Universitat de Lleida).• Membre del grup de recerca Essence: Data Engineering & Distributed Computing Systems des de 2022 (Universitat de Glasgow).• Computació distribuïda: <i>Cloud, Edge i Fog Computing</i>.• Camps d'aplicació: <i>Salut electrònica i Energia</i>.</div>

Introducció al curs

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
└─ Introducció al curs

Introducció al curs

Benvinguts a Sistemes Operatius

```
char text[] = "Aquesta NO és una assignatura\n\ncentrada en la programació\n\nerò programarem MOLT!\n";
ssize_t bytes = write(1, text, sizeof(text) - 1);

text="Aquesta assignatura NO és un MONÒLEG."
echo $text
```

Vull sessions interactives, participatives,... pregunteu, interrompeu-me...

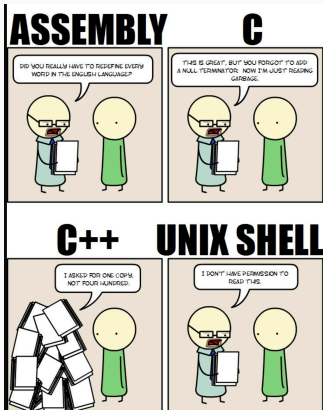


Figura 1: Comparativa de llenguatges

2024-09-30



- Determinar les característiques funcionals i el disseny dels elements que conformen un **Sistema Operatiu**.

- Determinar les característiques funcionals i el disseny dels elements que conformen un **Sistema Operatiu**.
- Analitzar la importància de cadascun dels mòduls que integren un **Sistema Operatiu**.

- Determinar les característiques funcionals i el disseny dels elements que conformen un **Sistema Operatiu**.
- Analitzar la importància de cadascun dels mòduls que integren un **Sistema Operatiu**.

- Determinar les característiques funcionals i el disseny dels elements que conformen un **Sistema Operatiu**.
- Analitzar la importància de cadascun dels mòduls que integren un **Sistema Operatiu**.
- Identificar els diferents serveis que proporciona el **Sistema Operatiu** en l'àmbit d'usuaris i d'aplicacions.

- Determinar les característiques funcionals i el disseny dels elements que conformen un **Sistema Operatiu**.
- Analitzar la importància de cadascun dels mòduls que integren un **Sistema Operatiu**.
- Identificar els diferents serveis que proporciona el **Sistema Operatiu** en l'àmbit d'usuaris i d'aplicacions.
- Utilitzar de forma eficient els serveis proporcionats pel **Sistema Operatiu** per al disseny i desenvolupament d'aplicacions informàtiques.

- Determinar les característiques funcionals i el disseny dels elements que conformen un **Sistema Operatiu**.
- Analitzar la importància de cadascun dels mòduls que integren un **Sistema Operatiu**.
- Identificar els diferents serveis que proporciona el **Sistema Operatiu** en l'àmbit d'usuaris i d'aplicacions.
- Utilitzar de forma eficient els serveis proporcionats pel **Sistema Operatiu** per al disseny i desenvolupament d'aplicacions informàtiques.

- Determinar les característiques funcionals i el disseny dels elements que conformen un **Sistema Operatiu**.
- Analitzar la importància de cadascun dels mòduls que integren un **Sistema Operatiu**.
- Identificar els diferents serveis que proporciona el **Sistema Operatiu** en l'àmbit d'usuaris i d'aplicacions.
- Utilitzar de forma eficient els serveis proporcionats pel **Sistema Operatiu** per al disseny i desenvolupament d'aplicacions informàtiques.
- Analitzar críticament les característiques i el funcionament de les polítiques que integren un **Sistema Operatiu**.

- Determinar les característiques funcionals i el disseny dels elements que conformen un **Sistema Operatiu**.
- Analitzar la importància de cadascun dels mòduls que integren un **Sistema Operatiu**.
- Identificar els diferents serveis que proporciona el **Sistema Operatiu** en l'àmbit d'usuaris i d'aplicacions.
- Utilitzar de forma eficient els serveis proporcionats pel **Sistema Operatiu** per al disseny i desenvolupament d'aplicacions informàtiques.
- Analitzar críticament les característiques i el funcionament de les polítiques que integren un **Sistema Operatiu**.

- Determinar les característiques funcionals i el disseny dels elements que conformen un **Sistema Operatiu**.
- Analitzar la importància de cadascun dels mòduls que integren un **Sistema Operatiu**.
- Identificar els diferents serveis que proporciona el **Sistema Operatiu** en l'àmbit d'usuaris i d'aplicacions.
- Utilitzar de forma eficient els serveis proporcionats pel **Sistema Operatiu** per al disseny i desenvolupament d'aplicacions informàtiques.
- Analitzar críticament les característiques i el funcionament de les polítiques que integren un **Sistema Operatiu**.
- Comparar de forma crítica els diferents mecanismes de gestió de la memòria.

- Determinar les característiques funcionals i el disseny dels elements que conformen un **Sistema Operatiu**.
- Analitzar la importància de cadascun dels mòduls que integren un **Sistema Operatiu**.
- Identificar els diferents serveis que proporciona el **Sistema Operatiu** en l'àmbit d'usuaris i d'aplicacions.
- Utilitzar de forma eficient els serveis proporcionats pel **Sistema Operatiu** per al disseny i desenvolupament d'aplicacions informàtiques.
- Analitzar críticament les característiques i el funcionament de les polítiques que integren un **Sistema Operatiu**.
- Comparar de forma crítica els diferents mecanismes de gestió de la memòria.

Objectius (Específics)

- Conèixer els sistemes operatius Unix/Linux (en concret **Debian**).

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció al curs

└─ Objectius (Específics)

2024-09-30

Objectius (Específics)

• Conèixer els sistemes operatius Unix/Linux (en concret **Debian**).

Objectius (Específics)

- Conèixer els sistemes operatius Unix/Linux (en concret **Debian**).
- Experimentar la programació en l'àmbit de sistema.

2024-09-30

- Unitat 0 · Sistemes Operatius (SO)
 - Introducció al curs
 - Objectius (Específics)

- Conèixer els sistemes operatius Unix/Linux (en concret **Debian**).
- Experimentar la programació en l'àmbit de sistema.

Objectius (Específics)

- Conèixer els sistemes operatius Unix/Linux (en concret **Debian**).
- Experimentar la programació en l'àmbit de sistema.
- Interioritza les bases de programació C per millora les vostres habilitats en altres llenguatges.

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció al curs

└─ Objectius (Específics)

- Conèixer els sistemes operatius Unix/Linux (en concret **Debian**).
- Experimentar la programació en l'àmbit de sistema.
- Interioritza les bases de programació C per millora les vostres habilitats en altres llenguatges.

- Introducció als Sistemes Operatius.

2024-09-30

- Unitat 0 · Sistemes Operatius (SO)
 - └ Introducció al curs
 - └ Temari

- Introducció als Sistemes Operatius.

- Introducció als Sistemes Operatius.
- Estructura dels Sistemes Operatius.

2024-09-30

- Unitat 0 · Sistemes Operatius (SO)
 - └ Introducció al curs
 - └ Temari

- Introducció als Sistemes Operatius.
- Estructura dels Sistemes Operatius.

- Introducció als Sistemes Operatius.
- Estructura dels Sistemes Operatius.
- Gestió i comunicació de processos i threads.

- Introducció als Sistemes Operatius.
- Estructura dels Sistemes Operatius.
- Gestió i comunicació de processos i threads.

- Introducció als Sistemes Operatius.
- Estructura dels Sistemes Operatius.
- Gestió i comunicació de processos i threads.
- Sincronització i Planificador de tasques.

- Introducció als Sistemes Operatius.
- Estructura dels Sistemes Operatius.
- Gestió i comunicació de processos i threads.
- Sincronització i Planificador de tasques.

- Introducció als Sistemes Operatius.
- Estructura dels Sistemes Operatius.
- Gestió i comunicació de processos i threads.
- Sincronització i Planificador de tasques.
- Interbloqueig.

- Introducció als Sistemes Operatius.
- Estructura dels Sistemes Operatius.
- Gestió i comunicació de processos i threads.
- Sincronització i Planificador de tasques.
- Interbloqueig.

- Introducció als Sistemes Operatius.
- Estructura dels Sistemes Operatius.
- Gestió i comunicació de processos i threads.
- Sincronització i Planificador de tasques.
- Interbloqueig.
- Gestió de Memòria.

- Introducció als Sistemes Operatius.
- Estructura dels Sistemes Operatius.
- Gestió i comunicació de processos i threads.
- Sincronització i Planificador de tasques.
- Interbloqueig.
- Gestió de Memòria.

- Introducció als Sistemes Operatius.
- Estructura dels Sistemes Operatius.
- Gestió i comunicació de processos i threads.
- Sincronització i Planificador de tasques.
- Interbloqueig.
- Gestió de Memòria.
- Scripting.

- Introducció als Sistemes Operatius.
- Estructura dels Sistemes Operatius.
- Gestió i comunicació de processos i threads.
- Sincronització i Planificador de tasques.
- Interbloqueig.
- Gestió de Memòria.
- Scripting.

- Introducció als Sistemes Operatius.
- Estructura dels Sistemes Operatius.
- Gestió i comunicació de processos i threads.
- Sincronització i Planificador de tasques.
- Interbloqueig.
- Gestió de Memòria.
- Scripting.

Propòsit

Els estudiants aprendreu a desenvolupar programari a nivell de sistema en el llenguatge de programació C comprnent al mateix temps com funcionen els **Sistemes Operatius** basats en **Unix**.

Temari
<ul style="list-style-type: none">• Introducció als Sistemes Operatius.• Estructura dels Sistemes Operatius.• Gestió i comunicació de processos i threads.• Sincronització i Planificador de tasques.• Interbloqueig.• Gestió de Memòria.• Scripting.
<p>Propòsit</p> <p>Els estudiants aprendreu a desenvolupar programari a nivell de sistema en el llenguatge de programació C comprnent al mateix temps com funcionen els Sistemes Operatius basats en Unix.</p>

Basada en ➡ Aprenentatge **SIGNIFICATIU**, **GUIAT** i **ACTIU**.

2024-09-30

- Unitat 0 · Sistemes Operatius (SO)
 - Introducció al curs
 - Metodologia

Basada en ➡ Aprenentatge **SIGNIFICATIU**, **GUIAT** i **ACTIU**.

- **Sessions teòriques:** S'introdueixen continguts teòrics de l'assignatura i als estudiants, i també s'hi discuteixen les implicacions pràctiques.
 - Diapositives (pdf i html).
 - Vídeos (youtube).

- **Sessions teòriques:** S'introdueixen continguts teòrics de l'assignatura i als estudiants, i també s'hi discuteixen les implicacions pràctiques.
 - Diapositives (pdf i html).
 - Vídeos (youtube).

Basada en ➡ Aprenentatge **SIGNIFICATIU**, **GUIAT** i **ACTIU**.

- **Sessions teòriques:** S'introdueixen continguts teòrics de l'assignatura i als estudiants, i també s'hi discuteixen les implicacions pràctiques.
 - Diapositives (pdf i html).
 - Vídeos (youtube).
- **Sessions pràctiques:** sessions autoguiades de laboratori, sessions de *live coding* o resolució de problemes.

Basada en ➡ Aprenentatge **SIGNIFICATIU**, **GUIAT** i **ACTIU**.

- **Sessions teòriques:** S'introdueixen continguts teòrics de l'assignatura i als estudiants, i també s'hi discuteixen les implicacions pràctiques.
 - Diapositives (pdf i html).
 - Vídeos (youtube).
- **Sessions pràctiques:** sessions autoguiades de laboratori, sessions de *live coding* o resolució de problemes.

Basada en ➡ Aprenentatge **SIGNIFICATIU**, **GUIAT** i **ACTIU**.

- **Sessions teòriques:** S'introdueixen continguts teòrics de l'assignatura i als estudiants, i també s'hi discuteixen les implicacions pràctiques.
 - Diapositives (pdf i html).
 - Vídeos (youtube).
- **Sessions pràctiques:** sessions autoguiades de laboratori, sessions de *live coding* o resolució de problemes.
- **Treball autònom:** els estudiants han d'aplicar els coneixements adquirits a les sessions teòriques i pràctiques per acabar els problemes, laboratoris i projectes proposats. A més, han de realitzar les lectures dels apunts i consultar la bibliografia recomanada.

Basada en ➡ Aprenentatge **SIGNIFICATIU**, **GUIAT** i **ACTIU**.

- **Sessions teòriques:** S'introdueixen continguts teòrics de l'assignatura i als estudiants, i també s'hi discuteixen les implicacions pràctiques.
 - Diapositives (pdf i html).
 - Vídeos (youtube).
- **Sessions pràctiques:** sessions autoguiades de laboratori, sessions de *live coding* o resolució de problemes.
- **Treball autònom:** els estudiants han d'aplicar els coneixements adquirits a les sessions teòriques i pràctiques per acabar els problemes, laboratoris i projectes proposats. A més, han de realitzar les lectures dels apunts i consultar la bibliografia recomanada.

Basada en ➡ Aprenentatge **SIGNIFICATIU**, **GUIAT** i **ACTIU**.

- **Sessions teòriques:** S'introdueixen continguts teòrics de l'assignatura i als estudiants, i també s'hi discuteixen les implicacions pràctiques.
 - Diapositives (pdf i html).
 - Vídeos (youtube).
- **Sessions pràctiques:** sessions autoguiades de laboratori, sessions de *live coding* o resolució de problemes.
- **Treball autònom:** els estudiants han d'aplicar els coneixements adquirits a les sessions teòriques i pràctiques per acabar els problemes, laboratoris i projectes proposats. A més, han de realitzar les lectures dels apunts i consultar la bibliografia recomanada.

Les sessions teòriques i pràctiques estan combinades en les sessions de 3h de durada.

Basada en **AP** Aprenentatge **SIGNIFICATIU**, **GUIAT** i **ACTIU**.

- **Sessions teòriques:** S'introdueixen continguts teòrics de l'assignatura i als estudiants, i també s'hi discuteixen les implicacions pràctiques.
 - Diapositives (pdf i html).
 - Vídeos (youtube).
- **Sessions pràctiques:** sessions autoguiades de laboratori, sessions de *live coding* o resolució de problemes.
- **Treball autònom:** els estudiants han d'aplicar els coneixements adquirits a les sessions teòriques i pràctiques per acabar els problemes, laboratoris i projectes proposats. A més, han de realitzar les lectures dels apunts i consultar la bibliografia recomanada.

Les sessions teòriques i pràctiques estan combinades en les sessions de 3h de durada.

Criteris d’Avaluació

Acr.	Activitat d’avaluació	Pes	Nota mínima	En grup	Recuperable
E1	Primer Parcial	42,5%	NO	NO	SI
E2	Segon Parcial	42,5%	NO	NO	SI
P	Projecte	10%	NO	$SI \leq 2$	NO
P1	Projecte 1	5%	NO	$SI \leq 2$	NO
P2	Projecte 2	5%	NO	$SI \leq 2$	NO
Part	Seguiment i Participació	5%	NO	NO	NO

Introducció al curs

Avaluació

Criteris d’Avaluació					
Acr.	Activitat d’avaluació	Pes	Nota mínima	En grup	Recuperable
E1	Primer Parcial	42,5%	NO	NO	SI
E2	Segon Parcial	42,5%	NO	NO	SI
P	Projecte	10%	NO	$SI \leq 2$	NO
P1	Projecte 1	5%	NO	$SI \leq 2$	NO
P2	Projecte 2	5%	NO	$SI \leq 2$	NO
Part	Seguiment i Participació	5%	NO	NO	NO

2024-09-30

Criteris d’Avaluació

Acr.	Activitat d’avaluació	Pes	Nota mínima	En grup	Recuperable
E1	Primer Parcial	42,5%	NO	NO	SI
E2	Segon Parcial	42,5%	NO	NO	SI
P	Projecte	10%	NO	SI ≤ 2	NO
P1	Projecte 1	5%	NO	SI ≤ 2	NO
P2	Projecte 2	5%	NO	SI ≤ 2	NO
Part	Seguiment i Participació	5%	NO	NO	NO

Extra

- 1. Manteniment de notes setmanals (fins a 10%).
- 2. Correcció, detecció i millora dels materials (fins a 10%).

Introducció al curs

Avaluació

2024-09-30

Criteris d’Avaluació					
Acr.	Activitat d’avaluació	Pes	Nota mínima	En grup	Recuperable
E1	Primer Parcial	42,5%	NO	NO	SI
E2	Segon Parcial	42,5%	NO	NO	SI
P	Projecte	10%	NO	SI ≤ 2	NO
P1	Projecte 1	5%	NO	SI ≤ 2	NO
P2	Projecte 2	5%	NO	SI ≤ 2	NO
Part	Seguiment i Participació	5%	NO	NO	NO
Extra					
1. Manteniment de notes setmanals (fins a 10%).					
2. Correcció, detecció i millora dels materials (fins a 10%).					

Extra: Manteniment de Notes Setmanals

- **Objectiu:** Fomentar la reflexió i l'autoavaluació dels continguts treballats a classe.
- **Metodologia:** Cada setmana, els estudiants han de fer un commit al seu repositori d'apunts amb les notes de les sessions de teoria i pràctiques.
- **Avaluació:** Es valorarà la qualitat de les notes, la seva coherència i la seva relació amb els continguts treballats, així com les reflexions i opinions personals incloses.

Unitat 0 · Sistemes Operatius (SO)

Introducció al curs

Extra: Manteniment de Notes Setmanals

2024-09-30

Extra: Manteniment de Notes Setmanals

- **Objectiu:** Fomentar la reflexió i l'autoavaluació dels continguts treballats a classe.
- **Metodologia:** Cada setmana, els estudiants han de fer un commit al seu repositori d'apunts amb les notes de les sessions de teoria i pràctiques.
- **Avaluació:** Es valorarà la qualitat de les notes, la seva coherència i la seva relació amb els continguts treballats, així com les reflexions i opinions personals incloses.

- **Objectiu:** Fomentar la reflexió i l'autoavaluació dels continguts treballats a classe.
- **Metodologia:** Cada setmana, els estudiants han de fer un commit al seu repositori d'apunts amb les notes de les sessions de teoria i pràctiques.
- **Avaluació:** Es valorarà la qualitat de les notes, la seva coherència i la seva relació amb els continguts treballats, així com les reflexions i opinions personals incloses.

Exemple de Notes

```
## Setmana X

# Resum teòric

# Exemples pràctics

# Dubtes i preguntes

# Reflexions personals
```

2024-09-30

• **Objectiu:** Fomentar la reflexió i l'autoavaluació dels continguts treballats a classe.

• **Metodologia:** Cada setmana, els estudiants han de fer un commit al seu repositori d'apunts amb les notes de les sessions de teoria i pràctiques.

• **Avaluació:** Es valorarà la qualitat de les notes, la seva coherència i la seva relació amb els continguts treballats, així com les reflexions i opinions personals incloses.

Exemple de Notes

```
## Setmana X

# Resum teòric

# Exemples pràctics

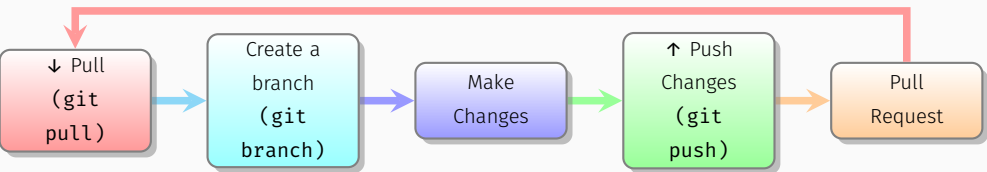
# Dubtes i preguntes

# Reflexions personals
```


Els materials de l'assignatura segueixen la metodologia de *Open Source*. Això vol dir que qualsevol estudiant pot detectar errors, millorar els materials o afegir continguts nous. Per tant, cada estudiant pot fer *fork* del repositori de l'assignatura i fer *pull request* amb les seves correccions i millores.

Extra: Correcció, Detecció i Millora dels Materials

Els materials de l'assignatura segueixen la metodologia de *Open Source*. Això vol dir que qualsevol estudiant pot detectar errors, millorar els materials o afegir continguts nous. Per tant, cada estudiant pot fer *fork* del repositori de l'assignatura i fer *pull request* amb les seves correccions i millores.



2024-09-30

Unitat 0 · Sistemes Operatius (SO)

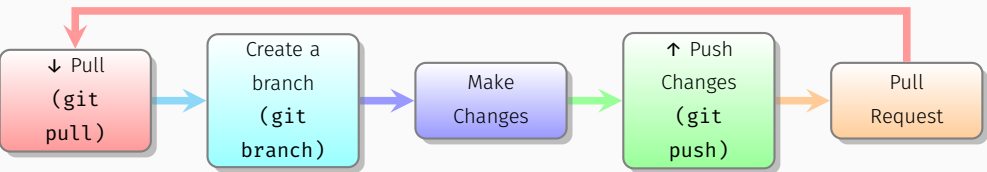
└─ Introducció al curs

└─ Extra: Correcció, Detecció i Millora dels Materials



Extra: Correcció, Detecció i Millora dels Materials

Els materials de l'assignatura segueixen la metodologia de *Open Source*. Això vol dir que qualsevol estudiant pot detectar errors, millorar els materials o afegir continguts nous. Per tant, cada estudiant pot fer *fork* del repositori de l'assignatura i fer *pull request* amb les seves correccions i millores.



- **Branca Main:** Reservada per estar sincronitzada amb els materials oficials.
- **Branques de treball:** Cada estudiant pot crear una branca per fer canvis i millorar els materials.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció al curs

└─ Extra: Correcció, Detecció i Millora dels Materials

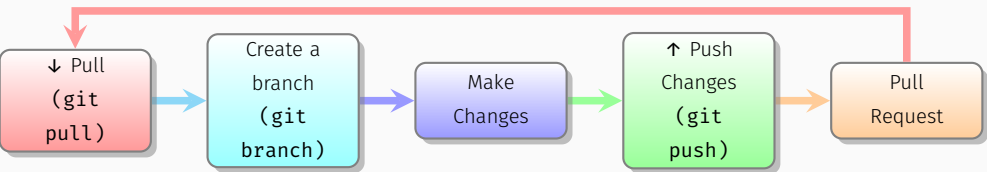
Extra: Correcció, Detecció i Millora dels Materials

Els materials de l'assignatura segueixen la metodologia de Open Source. Això vol dir que qualsevol estudiant pot detectar errors, millorar els materials o afegir continguts nous. Per tant, cada estudiant pot fer fork del repositori de l'assignatura i fer pull request amb les seves correccions i millores.

• **Branca Main:** Reservada per estar sincronitzada amb els materials oficials.
• **Branques de treball:** Cada estudiant pot crear una branca per fer canvis i millorar els materials.

Extra: Correcció, Detecció i Millora dels Materials

Els materials de l'assignatura segueixen la metodologia de *Open Source*. Això vol dir que qualsevol estudiant pot detectar errors, millorar els materials o afegir continguts nous. Per tant, cada estudiant pot fer *fork* del repositori de l'assignatura i fer *pull request* amb les seves correccions i millores.



- **Branca Main:** Reservada per estar sincronitzada amb els materials oficials.
- **Branques de treball:** Cada estudiant pot crear una branca per fer canvis i millorar els materials.

S'avaluarà la quantitat i qualitat de les correccions, deteccions i millores realitzades en els materials de l'assignatura.

Extra: Correcció, Detecció i Millora dels Materials

Els materials de l'assignatura segueixen la metodologia de Open Source. Això vol dir que qualsevol estudiant pot detectar errors, millorar els materials o afegir continguts nous. Per tant, cada estudiant pot fer fork del repositori de l'assignatura i fer pull request amb les seves correccions i millores.

• **Branca Main:** Reservada per estar sincronitzada amb els materials oficials.
• **Branques de treball:** Cada estudiant pot crear una branca per fer canvis i millorar els materials.

S'avaluarà la quantitat i qualitat de les correccions, deteccions i millores realitzades en els materials de l'assignatura.

- L'assistència a les sessions de teoria i pràctiques és obligatòria però altament *recomanable*.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció al curs

└─ Normativa del curs (I) - *Assistència i Seguiment*

- L'assistència a les sessions de teoria i pràctiques és obligatòria però altament *recomanable*.
- **Seguiment i Participació:** Es valorarà la participació activa a les sessions de teoria i pràctiques, així com la realització i entrega dels exercicis i problemes proposats.

- L'assistència a les sessions de teoria i pràctiques és obligatòria però altament recomanable.
- **Seguiment i Participació:** Es valorarà la participació activa a les sessions de teoria i pràctiques, així com la realització i entrega dels exercicis i problemes proposats.

- L'assistència a les sessions de teoria i pràctiques és obligatòria però altament *recomanable*.
- **Seguiment i Participació:** Es valorarà la participació activa a les sessions de teoria i pràctiques, així com la realització i entrega dels exercicis i problemes proposats.
- Els projectes pràctics no són obligatoris, i es poden realitzar en grup de fins a 3 persones.

- L'assistència a les sessions de teoria i pràctiques és obligatòria però altament *recomanable*.
- **Seguiment i Participació:** Es valorarà la participació activa a les sessions de teoria i pràctiques, així com la realització i entrega dels exercicis i problemes proposats.
- Els projectes pràctics no són obligatoris, i es poden realitzar en grup de fins a 3 persones.
- Es demana que els estudiants arribin puntuals a les sessions. L'entrada tardana pot interrompre la dinàmica de la classe i el treball dels companys.

- Cada estudiant és responsable de la seva pròpia feina i de com gestiona el seu temps. L'assistència a classe no és obligatòria, però és altament recomanable per garantir el seguiment adequat de l'assignatura.

- Cada estudiant és responsable de la seva pròpia feina i de com gestiona el seu temps. L'assistència a classe no és obligatòria, però és altament recomanable per garantir el seguiment adequat de l'assignatura.
- Durant les sessions de laboratori, es fomenta el treball en grup, sempre de manera ordenada i respectuosa envers els companys.

- Cada estudiant és responsable de la seva pròpia feina i de com gestiona el seu temps. L'assistència a classe no és obligatòria, però és altament recomanable per garantir el seguiment adequat de l'assignatura.
- Durant les sessions de laboratori, es fomenta el treball en grup, sempre de manera ordenada i respectuosa envers els companys.
- Cada alumne ha de presentar evidències pròpies del treball realitzat, encara que es col·labori en grup.

- Els exàmens són escrits i no està permès l'ús d'ordinadors. Es permetrà una fulla manuscrita mida A4 amb apunts, notes, etc.

- Els exàmens són escrits i no està permès l'ús d'ordinadors. Es permetrà una fulla manuscrita mida A4 amb apunts, notes, etc.
- Els parcials avaluaran els coneixements teòrics i pràctics de l'assignatura.

- Els exàmens són escrits i no està permès l'ús d'ordinadors. Es permetrà una fulla manuscrita mida A4 amb apunts, notes, etc.
- Els parcials avaluaran els coneixements teòrics i pràctics de l'assignatura.

- Els exàmens són escrits i no està permès l'ús d'ordinadors. Es permetrà una fulla manuscrita mida A4 amb apunts, notes, etc.
- Els parcials avaluaran els coneixements teòrics i pràctics de l'assignatura.
- L'ús d'eines d'intel·ligència artificial està permès durant el curs, sempre que no es limiti a un simple copiar i enganxar. Heu de justificar les vostres respostes amb reflexions i opinions personals.

- Ordinador portàtil amb connexió a Internet.
 - Es recomana portar el vostre propi ordinador per cursar l'assignatura.
 - Si no disposeu d'ordinador, podreu utilitzar els ordinadors de la classe durant les sessions de laboratori.

- Ordinador portàtil amb connexió a Internet.
 - Es recomana portar el vostre propi ordinador per cursar l'assignatura.
 - Si no disposeu d'ordinador, podreu utilitzar els ordinadors de la classe durant les sessions de laboratori.
- Distribucions Linux (Debian)
 - És obligatori utilitzar la distribució Debian per a les pràctiques de laboratori.

- Ordinador portàtil amb connexió a Internet.
 - Es recomana portar el vostre propi ordinador per cursar l'assignatura.
 - Si no disposeu d'ordinador, podreu utilitzar els ordinadors de la classe durant les sessions de laboratori.
- Distribucions Linux (Debian)
 - És obligatori utilitzar la distribució Debian per a les pràctiques de laboratori.
- Software de virtualització (VMWare)
 - Es recomana utilitzar VMWare, però podeu optar per altres opcions de virtualització. Heu de ser capaços d'adaptar el material al vostre programari de virtualització.

- Ordinador portàtil amb connexió a Internet.
 - Es recomana portar el vostre propi ordinador per cursar l'assignatura.
 - Si no disposeu d'ordinador, podreu utilitzar els ordinadors de la classe durant les sessions de laboratori.
- Distribucions Linux (Debian)
 - És obligatori utilitzar la distribució Debian per a les pràctiques de laboratori.
- Software de virtualització (VMWare)
 - Es recomana utilitzar VMWare, però podeu optar per altres opcions de virtualització. Heu de ser capaços d'adaptar el material al vostre programari de virtualització.

Llenguatges de programació i compiladors

- C
- GCC

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
└─ Introducció al curs

└─ Programari necessari

Programari necessari

Llenguatges de programació i compiladors

- C
- GCC

Llenguatges de programació i compiladors

- C
- GCC

Eines de control de versions

- Git
- Github

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció al curs

└─ Programari necessari

Programari necessari

Llenguatges de programació i compiladors

• C

• GCC

Eines de control de versions

• Git

• Github

Llenguatges de programació i compiladors

- C
- GCC

Eines de control de versions

- Git
- Github

IDE

- Visual Studio (Recomanat)
- Vi, Vim, NeoVim, Emacs, CLion, Eclipse ...

Llenguatges de programació i compiladors

- C
- GCC

Eines de control de versions

- Git
- Github

IDE

- Visual Studio (Recomanat)
- Vi, Vim, NeoVim, Emacs, CLion, Eclipse ...

- Apunts de l'assignatura i materials complementaris proporcionats pel professor al campus virtual.

- Apunts de l'assignatura i materials complementaris proporcionats pel professor al campus virtual.
- Laboratoris i activitats pràctiques disponibles al repositori de l'assignatura.

- Apunts de l'assignatura i materials complementaris proporcionats pel professor al campus virtual.
- Laboratoris i activitats pràctiques disponibles al repositori de l'assignatura.
- Documentació oficial de les eines i tecnologies utilitzades.

- Apunts de l'assignatura i materials complementaris proporcionats pel professor al campus virtual.
- Laboratoris i activitats pràctiques disponibles al repositori de l'assignatura.
- Documentació oficial de les eines i tecnologies utilitzades.
- Fòrums tècnics a la xarxa com *Stack Overflow*, *Reddit*, etc.

- Apunts de l'assignatura i materials complementaris proporcionats pel professor al campus virtual.
- Laboratoris i activitats pràctiques disponibles al repositori de l'assignatura.
- Documentació oficial de les eines i tecnologies utilitzades.
- Fòrums tècnics a la xarxa com *Stack Overflow*, *Reddit*, etc.
- Llibres de referència en Sistemes Operatius:
 - Operating System Concepts; Abraham Silberschatz
 - Modern Operating Systems; Andrew S. Tanenbaum

- Apunts de l'assignatura i materials complementaris proporcionats pel professor al campus virtual.
- Laboratoris i activitats pràctiques disponibles al repositori de l'assignatura.
- Documentació oficial de les eines i tecnologies utilitzades.
- Fòrums tècnics a la xarxa com *Stack Overflow*, *Reddit*, etc.
- Llibres de referència en Sistemes Operatius:
 - Operating System Concepts; Abraham Silberschatz
 - Modern Operating Systems; Andrew S. Tanenbaum

- DRY (Do not repeat yourself).

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció al curs

└─ Bones pràctiques

2024-09-30

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.

2024-09-30

- Unitat 0 · Sistemes Operatius (SO)
 - └ Introducció al curs
 - └ Bones pràctiques

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.
- No assumeixis res, prova-ho.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció al curs

└─ Bones pràctiques

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.
- No assumeixis res, prova-ho.

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.
- No assumeixis res, prova-ho.
- Utilitza assertions per prevenir l'impossible.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
└─ Introducció al curs
└─ Bones pràctiques

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.
- No assumeixis res, prova-ho.
- Utilitza assertions per prevenir l'impossible.

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.
- No assumeixis res, prova-ho.
- Utilitza assertions per prevenir l'impossible.
- Utilitza excepcions per problemes excepcionals.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció al curs

└─ Bones pràctiques

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.
- No assumeixis res, prova-ho.
- Utilitza assertions per prevenir l'impossible.
- Utilitza excepcions per problemes excepcionals.

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.
- No assumeixis res, prova-ho.
- Utilitza assertions per prevenir l'impossible.
- Utilitza excepcions per problemes excepcionals.
- Estima l'ordre de complexitat dels teus algorismes.

2024-09-30

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.
- No assumeixis res, prova-ho.
- Utilitza assertions per prevenir l'impossible.
- Utilitza excepcions per problemes excepcionals.
- Estima l'ordre de complexitat dels teus algorismes.

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.
- No assumeixis res, prova-ho.
- Utilitza assertions per prevenir l'impossible.
- Utilitza excepcions per problemes excepcionals.
- Estima l'ordre de complexitat dels teus algorismes.
- Utilitza patrons de disseny.

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.
- No assumeixis res, prova-ho.
- Utilitza assertions per prevenir l'impossible.
- Utilitza excepcions per problemes excepcionals.
- Estima l'ordre de complexitat dels teus algorismes.
- Utilitza patrons de disseny.

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.
- No assumeixis res, prova-ho.
- Utilitza assertions per prevenir l'impossible.
- Utilitza excepcions per problemes excepcionals.
- Estima l'ordre de complexitat dels teus algorismes.
- Utilitza patrons de disseny.
- Utilitza eines de control de versions.

- DRY (Do not repeat yourself).
- Codi fàcil de reutilitzar.
- Testeja aviat, testeja sovint, testeja de forma automàtica.
- No assumeixis res, prova-ho.
- Utilitza assertions per prevenir l'impossible.
- Utilitza excepcions per problemes excepcionals.
- Estima l'ordre de complexitat dels teus algorismes.
- Utilitza patrons de disseny.
- Utilitza eines de control de versions.

The Pragmatic Programmer, Andrew Hunt David Thomas

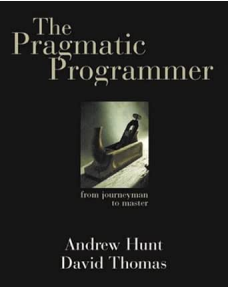


Figura 2: The Pragmatic Programmer

Clean Code, (A Handbook of Agile Software Craftsmanship),Robert C. Martin

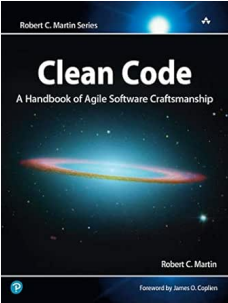


Figura 3: Clean Code

2024-09-30



Introducció a la temàtica

Què tenen en comú?

- Cotxe



Figura 4: Diagrama sobre els dispositius actuals.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
└ Introducció a la temàtica

└ Societat actual



Figura 4: Diagrama sobre els dispositius actuals.

Què tenen en comú?

- Cotxe
- Rellotge



Figura 4: Diagrama sobre els dispositius actuals.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
└ Introducció a la temàtica

└ Societat actual



Figura 4: Diagrama sobre els dispositius actuals.

Què tenen en comú?

- Cotxe
- Rellotge
- Portàtil



Figura 4: Diagrama sobre els dispositius actuals.

2024-09-30



Figura 4: Diagrama sobre els dispositius actuals.

Què tenen en comú?

- Cotxe
- Rellotge
- Portàtil
- PC



Figura 4: Diagrama sobre els dispositius actuals.

2024-09-30



Figura 4: Diagrama sobre els dispositius actuals.

Què tenen en comú?

- Cotxe
- Rellotge
- Portàtil
- PC
- Tablet



Figura 4: Diagrama sobre els dispositius actuals.

2024-09-30

Què tenen en comú?

- Cotxe
- Rellotge
- Portàtil
- PC
- Tablet
- Rentadora



Figura 4: Diagrama sobre els dispositius actuals.

2024-09-30

└ Introducció a la temàtica

└ Societat actual



Figura 4: Diagrama sobre els dispositius actuals.

Què tenen en comú?

- Cotxe
- Rellotge
- Portàtil
- PC
- Tablet
- Rentadora
- Nevera



Figura 4: Diagrama sobre els dispositius actuals.

2024-09-30

└ Introducció a la temàtica

└ Societat actual



Figura 4: Diagrama sobre els dispositius actuals.

Què tenen en comú?

- Cotxe
- Rellotge
- Portàtil
- PC
- Tablet
- Rentadora
- Nevera
- Televisió



Figura 4: Diagrama sobre els dispositius actuals.

2024-09-30

└ Introducció a la temàtica

└ Societat actual



Figura 4: Diagrama sobre els dispositius actuals.

Què tenen en comú?

- Cotxe
- Rellotge
- Portàtil
- PC
- Tablet
- Rentadora
- Nevera
- Televisió
- ...



Figura 4: Diagrama sobre els dispositius actuals.

2024-09-30



Figura 4: Diagrama sobre els dispositius actuals.



Avui en dia tothom parla de **IoT, BigData, Cloud, AI, Blockchain, Metavers**

- Microprocessador a tot arreu.

Casa connectada (Font: Extret de SemanticScholar)

Figura 5: Casa connectada (Font: Extret de SemanticScholar)

En la societat actual es parla constatant de noves tecnologies disruptives com **IoT, BigData, Cloud, AI, Blockchain, Metavers**,... Això fa que vivim en un món connectat on tots els dispositius són capaços de generar dades; per exemple, vivim envoltats de dispositius mòbils i sensors de tot tipus, també tenim servidors en centres de dades distribuïts per tot el món (inclús a sota el mar) i també serveis capaços de processar i emmagatzemar grans quantitats de dades i d'informació.

Per tant, com a societat tenim la capacitat d'interactuar amb el nostre entorn i ambient.

- Quants teniu un cotxe? Quants processadors creieu que té un cotxe?
 - Més de 50!
- Qui utilitza un smartphone? Quants processadors/cores té el vostre smartphone?
 - El meu un Poco X4 GT té un MediaTek Dimensity 8100 amb 8 nuclis!

Tots els dispositius que fem servir de forma quotidiana són sistemes paral·lels que funcionen i interactuen entre ells. Per tant, podem afirmar que el món és un gran sistema paral·lel.

Avui en dia tothom parla de IoT, BigData, Cloud, AI, Blockchain, Metavers

- Microprocessador a tot arreu.
- Xarxes i Connectivitat.

Casa connectada (Font: Extret de SemanticScholar)

Figura 5: Casa connectada (Font: Extret de SemanticScholar)

En la societat actual es parla constatant de noves tecnologies disruptives com IoT, BigData, Cloud, AI, Blockchain, Metavers,... Això fa que vivim en un món connectat on tots els dispositius són capaços de generar dades; per exemple, vivim envoltats de dispositius mòbils i sensors de tot tipus, també tenim servidors en centres de dades distribuïts per tot el món (inclús a sota el mar) i també serveis capaços de processar i emmagatzemar grans quantitats de dades i d’informació.

Per tant, com a societat tenim la capacitat d’interactuar amb el nostre entorn i ambient.

- Quants teniu un cotxe? Quants processadors creieu que té un cotxe?
 - Més de 50!
- Qui utilitza un smartphone? Quants processadors/cores té el vostre smartphone?
 - El meu un Poco X4 GT té un MediaTek Dimensity 8100 amb 8 nuclis!

Tots els dispositius que fem servir de forma quotidiana són sistemes paral·lels que funcionen i interactuen entre ells. Per tant, podem afirmar que el món és un gran sistema paral·lel.

Avui en dia tothom parla de IoT, BigData, Cloud, AI, Blockchain, Metavers

- Microprocessador a tot arreu.
- Xarxes i Connectivitat.
- Serveis escalables, confiables i segurs.

Casa connectada (Font: Extret de SemanticScholar)

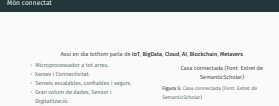
Figura 5: Casa connectada (Font: Extret de SemanticScholar)

En la societat actual es parla constatant de noves tecnologies disruptives com IoT, BigData, Cloud, AI, Blockchain, Metavers,... Això fa que vivim en un món connectat on tots els dispositius són capaços de generar dades; per exemple, vivim envoltats de dispositius mòbils i sensors de tot tipus, també tenim servidors en centres de dades distribuïts per tot el món (inclús a sota el mar) i també serveis capaços de processar i emmagatzemar grans quantitats de dades i d’informació.

Per tant, com a societat tenim la capacitat d’interactuar amb el nostre entorn i ambient.

- Quants teniu un cotxe? Quants processadors creieu que té un cotxe?
 - Més de 50!
- Qui utilitza un smartphone? Quants processadors/cores té el vostre smartphone?
 - El meu un Poco X4 GT té un MediaTek Dimensity 8100 amb 8 nuclis!

Tots els dispositius que fem servir de forma quotidiana són sistemes paral·lels que funcionen i interactuen entre ells. Per tant, podem afirmar que el món és un gran sistema paral·lel.



Avui en dia tothom parla de **IoT, BigData, Cloud, AI, Blockchain, Metavers**

- Microprocessador a tot arreu.
- Xarxes i Connectivitat.
- Serveis escalables, confiables i segurs.
- Gran volum de dades, Sensor i Digitalització.

Casa connectada (Font: Extret de SemanticScholar)

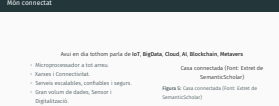
Figura 5: Casa connectada (Font: Extret de SemanticScholar)

En la societat actual es parla constatant de noves tecnologies disruptives com **IoT, BigData, Cloud, AI, Blockchain, Metavers**,... Això fa que vivim en un món connectat on tots els dispositius són capaços de generar dades; per exemple, vivim envoltats de dispositius mòbils i sensors de tot tipus, també tenim servidors en centres de dades distribuïts per tot el món (inclús a sota el mar) i també serveis capaços de processar i emmagatzemar grans quantitats de dades i d'informació.

Per tant, com a societat tenim la capacitat d'interactuar amb el nostre entorn i ambient.

- Quants teniu un cotxe? Quants processadors creieu que té un cotxe?
 - Més de 50!
- Qui utilitza un smartphone? Quants processadors/cores té el vostre smartphone?
 - El meu un Poco X4 GT té un MediaTek Dimensity 8100 amb 8 nuclis!

Tots els dispositius que fem servir de forma quotidiana són sistemes paral·lels que funcionen i interactuen entre ells. Per tant, podem afirmar que el món és un gran sistema paral·lel.



Avui en dia tothom parla de **IoT, BigData, Cloud, AI, Blockchain, Metavers**

- Microprocessador a tot arreu.
- Xarxes i Connectivitat.
- Serveis escalables, confiables i segurs.
- Gran volum de dades, Sensor i Digitalització.

Casa connectada (Font: Extret de SemanticScholar)

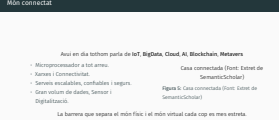
Figura 5: Casa connectada (Font: Extret de SemanticScholar)

En la societat actual es parla constatant de noves tecnologies disruptives com **IoT, BigData, Cloud, AI, Blockchain, Metavers**,... Això fa que vivim en un món connectat on tots els dispositius són capaços de generar dades; per exemple, vivim envoltats de dispositius mòbils i sensors de tot tipus, també tenim servidors en centres de dades distribuïts per tot el món (inclús a sota el mar) i també serveis capaços de processar i emmagatzemar grans quantitats de dades i d’informació.

Per tant, com a societat tenim la capacitat d’interactuar amb el nostre entorn i ambient.

- Quants teniu un cotxe? Quants processadors creieu que té un cotxe?
 - Més de 50!
- Qui utilitza un smartphone? Quants processadors/cores té el vostre smartphone?
 - El meu un Poco X4 GT té un MediaTek Dimensity 8100 amb 8 nuclis!

Tots els dispositius que fem servir de forma quotidiana són sistemes paral·lels que funcionen i interactuen entre ells. Per tant, podem afirmar que el món és un gran sistema paral·lel.



Avui en dia tothom parla de **IoT, BigData, Cloud, AI, Blockchain, Metavers**

- Microprocessador a tot arreu.
- Xarxes i Connectivitat.
- Serveis escalables, confiables i segurs.
- Gran volum de dades, Sensor i Digitalització.

Casa connectada (Font: Extret de SemanticScholar)

Figura 5: Casa connectada (Font: Extret de SemanticScholar)

La barrera que separa el món físic i el món virtual cada cop es mes estreta.

En la societat actual es parla constatant de noves tecnologies disruptives com **IoT, BigData, Cloud, AI, Blockchain, Metavers**,... Això fa que vivim en un món connectat on tots els dispositius són capaços de generar dades; per exemple, vivim envoltats de dispositius mòbils i sensors de tot tipus, també tenim servidors en centres de dades distribuïts per tot el món (inclús a sota el mar) i també serveis capaços de processar i emmagatzemar grans quantitats de dades i d’informació.

Per tant, com a societat tenim la capacitat d’interactuar amb el nostre entorn i ambient.

- Quants teniu un cotxe? Quants processadors creieu que té un cotxe?
 - Més de 50!
- Qui utilitza un smartphone? Quants processadors/cores té el vostre smartphone?
 - El meu un Poco X4 GT té un MediaTek Dimensity 8100 amb 8 nuclis!

Tots els dispositius que fem servir de forma quotidiana són sistemes paral·lels que funcionen i interactuen entre ells. Per tant, podem afirmar que el món és un gran sistema paral·lel.

Què tenen en comú aquests progressos tecnològics?



Figura 6: Diagrama sobre els dispositius actuals.

Una *interfície (Sistema Operatiu)* capaç d'integrar una gran diversitat de maquinari i programari i comunicar-se a través de la xarxa.

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què tenen en comú aquests progressos tecnològics?

Però, **que tenen en comú aquests progressos tecnològics?** Requereixen un cervell que permeti a tots els diferents aparells i màquines interactuar amb nosaltres, tant per generar, processar o guardar. Tots els dispositius i servidors requereixen un sistema operatiu que permeti maquinari i programari comunicar-se.

En aquest curs tractarem els cervells d'aquests sistemes i analitzarem de quines estructures podem dotar aquests sistemes perquè funcionin correctament i permeti que maquinari tan heterogeni sigui capaç de fer funcionar infinitat de programes i serveis ➡ per gestionar eficientment les dades i la informació del nostre entorn.

Per tant, a la pregunta inicial què tenen en comú? podem afirmar:

- Tenen en comú que tots són sistemes informàtics.
- Tots tenen un sistema operatiu.
- Tots es comuniquen per internet amb altres sistemes.



Què són les lleis de Moore i Bell?

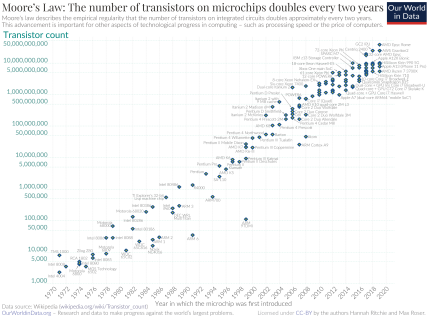


Figura 7: Llei de Moore

Unitat 0 · Sistemes Operatius (SO)

└ Introducció a la temàtica

└ Què són les lleis de Moore i Bell?



Figura 7: Llei de Moore

En aquestes dues figures podeu observar la llei de moore i la llei de bell. En la primera figura, es pot observar com la capacitat dels microprocessadors s’ha anat duplicant cada dos anys sense tenir en compte el cost. En la segona figura es pot observar com la mida i les classes de computadors han evolucionat cada 10 anys.

Venim de la dècada dels anys 60 i 70, on es van desenvolupar els primers microprocessadors i la informàtica personal estava en les seves primeres etapes. En aquesta època, es feien servir pocs processadors que eren compartits per moltes persones. Els avanços en la capacitat de processament eren més lents, i la tecnologia era més limitada.

Ara, en l’actualitat, cada persona fa servir molts processadors de manera habitual. Això es deu a l’augment de la potència de processament dels dispositius que utilitzem, com els telèfons intel·ligents i les tauletes, així com als ordinadors personals i altres tecnologies.

Pel que fa al futur, podem esperar que la tendència continuï amb l’augment de la capacitat de processament i la miniaturització de la tecnologia. Això pot tenir un impacte significatiu en molts àmbits de la vida, com la informàtica, la medicina, la intel·ligència artificial i molts altres. No obstant això, també s’han plantejat preocupacions sobre els límits físics d’aquest creixement i els impactes ambientals associats a l’augment de la producció tecnològica.

Què són les lleis de Moore i Bell?

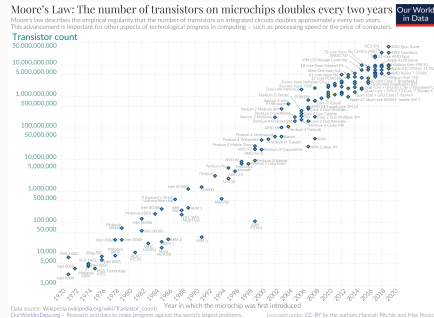


Figura 7: Llei de Moore

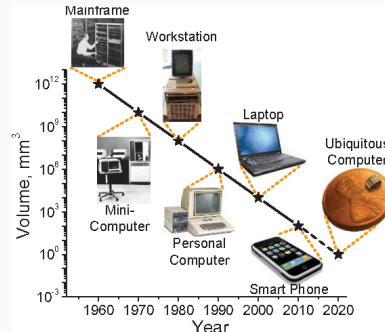


Figura 8: Llei de Bell

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

- └ Introducció a la temàtica
- └ Què són les lleis de Moore i Bell?



Figure 7: Llei de Moore



Figure 8: Llei de Bell

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

- └ Introducció a la temàtica
- └ Què són les lleis de Moore i Bell?



Figure 7: Llei de Moore



Figure 8: Llei de Bell

—Què són les lleis de Moore i Bell?

En aquestes dues figures podeu observar la llei de Moore i la llei de Bell. En la primera figura, es pot observar com la capacitat dels microprocessadors s'ha anat duplicant cada dos anys sense tenir en compte el cost. En la segona figura es pot observar com la mida i les classes de computadors han evolucionat cada 10 anys.

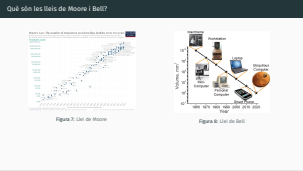
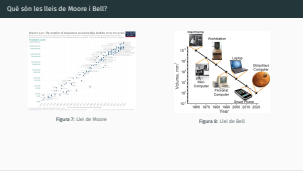
Venim de la dècada dels anys 60 i 70, on es van desenvolupar els primers microprocessadors i la informàtica personal estava en les seves primeres etapes. En aquesta època, es feien servir pocs processadors que eren compartits per moltes persones. Els avanços en la capacitat de processament eren més lents, i la tecnologia era més limitada.

Ara, en l'actualitat, cada persona fa servir molts processadors de manera habitual. Això es deu a l'augment de la potència de processament dels dispositius que utilitzem, com els telèfons intel·ligents i les tauletes, així com als ordinadors personals i altres tecnologies.

Pel que fa al futur, podem esperar que la tendència continuï amb l'augment de la capacitat de processament i la miniaturització de la tecnologia. Això pot tenir un impacte significatiu en molts àmbits de la vida, com la informàtica, la medicina, la intel·ligència artificial i molts altres. No obstant això, també s'han plantejat preocupacions sobre els límits físics d'aquest creixement i els impactes ambientals associats a l'augment de la producció tecnològica.

Pel que fa al futur, podem esperar que la tendència continuï amb l'augment de la capacitat de processament i la miniaturització de la tecnologia. Això pot tenir un impacte significatiu en molts àmbits de la vida, com la informàtica, la medicina, la intel·ligència artificial i molts altres. No obstant això, també s'han plantejat preocupacions sobre els límits físics d'aquest creixement i els impactes ambientals associats a l'augment de la producció tecnològica.

Pel que fa al futur, podem esperar que la tendència continuï amb l'augment de la capacitat de processament i la miniaturització de la tecnologia. Això pot tenir un impacte significatiu en molts àmbits de la vida, com la informàtica, la medicina, la intel·ligència artificial i molts altres. No obstant això, també s'han plantejat preocupacions sobre els límits físics d'aquest creixement i els impactes ambientals associats a l'augment de la producció tecnològica.



Què són les lleis de Moore i Bell?

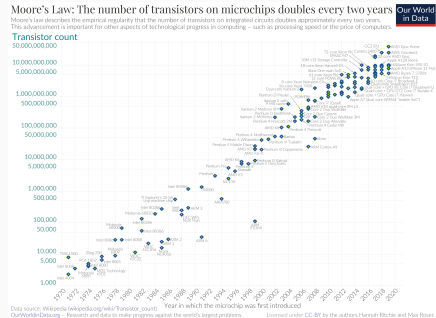


Figura 7: Llei de Moore

Venim de pocs processadors fets servir per a moltes persones (anys 60-70) i ara cada persona fa servir molts processadors. Com ha canviat la tendència... on ens portarà el futur?

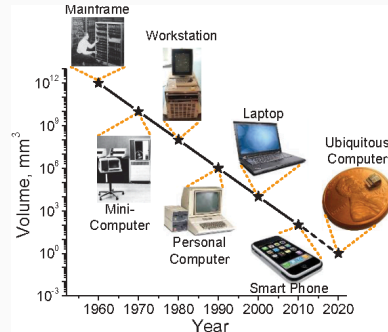


Figura 8: Llei de Bell

2024-09-30

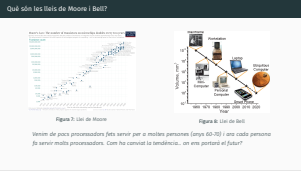
Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què són les lleis de Moore i Bell?

- Introducció a la temàtica

—Què són les lleis de Moore i Bell?



En aquestes dues figures podeu observar la llei de moore i la llei de bell. En la primera figura, es pot observar com la capacitat dels microprocessadors s'ha anat duplicant cada dos anys sense tenir en compte el cost. En la segona figura es pot observar com la mida i les classes de computadors han evolucionat cada 10 anys.

Venim de la dècada dels anys 60 i 70, on es van desenvolupar els primers microprocessadors i la informàtica personal estava en les seves primeres etapes. En aquesta època, es feien servir pocs processadors que eren compartits per moltes persones. Els avanços en la capacitat de processament eren més lents, i la tecnologia era més limitada.

Ara, en l'actualitat, cada persona fa servir molts processadors de manera habitual. Això es deu a l'augment de la potència de processament dels dispositius que utilitzem, com els telèfons intel·ligents i les tauletes, així com als ordinadors personals i altres tecnologies.

Pel que fa al futur, podem esperar que la tendència continuï amb l'augment de la capacitat de processament i la miniaturització de la tecnologia. Això pot tenir un impacte significatiu en molts àmbits de la vida, com la informàtica, la medicina, la intel·ligència artificial i molts altres. No obstant això, també s'han plantejat preocupacions sobre els límits físics d'aquest creixement i els impactes ambientals associats a l'augment de la producció tecnològica.

Què és un sistema informàtic?

Un **sistema informàtic** és la interconnexió d'elements de maquinari per exemple **d'1 o més CPU, memòria** i components E/S ... Amb la finalitat d'executar programes i accions (en sèrie o de forma concurrent) per **1 o múltiples usuaris**.

Unitat 0 · Sistemes Operatius (SO)

2024-09-30

└ Introducció a la temàtica

└ Què és un sistema informàtic?

Un **sistema informàtic** és la interconnexió d'elements de maquinari per exemple **d'1 o més CPU, memòria** i components E/S ... Amb la finalitat d'executar programes i accions (en sèrie o de forma concurrent) per **1 o múltiples usuaris**.

Per què necessitem la concurrència?

Com heu vist el mon real actua com un gran sistema paral·lel. Per tant, tots els sistemes de temps real son inherentment concurrents i lògicament els nostres sisteme informàtics també han de poder actuar en paral·lel. En la figura, podeu observar com una tasca es descomposa en diferents parts per poder aprofitar els diferents processadors i nuclis de cada processador.

L'execució **concurrent** recull un conjunt de tècniques informàtiques usades per representar i gestionar el paral·lelisme i les eines de sincronització i comunicació entre programes.

Què és un sistema informàtic?
Un **sistema informàtic** és la interconnexió d'elements de maquinari per exemple **d'1 o més CPU, memòria** i components E/S ... Amb la finalitat d'executar programes i accions (en sèrie o de forma concurrent) per **1 o múltiples usuaris**.

Què és un sistema informàtic?

Un **sistema informàtic** és la interconnexió d'elements de maquinari per exemple **d'1 o més CPU, memòria** i components E/S ... Amb la finalitat d'executar programes i accions (en sèrie o de forma concurrent) per **1 o múltiples usuaris**.

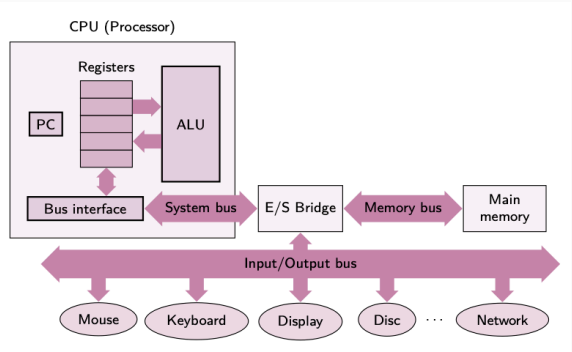


Figura 10: Esquema d'un sistema informàtic

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què és un sistema informàtic?

Què és un sistema informàtic?

Un **sistema informàtic** és la interconnexió d'elements de maquinari per exemple **d'1 o més CPU, memòria** i components E/S ... Amb la finalitat d'executar programes i accions (en sèrie o de forma concurrent) per **1 o múltiples usuaris**.

Figura 10: Esquema d'un sistema informàtic

Un **sistema informàtic** és la interconnexió d'elements de maquinari per exemple **d'1 o més CPU, memòria** i components E/S ... Amb la finalitat d'executar programes i accions (en sèrie o de forma concurrent) per **1 o múltiples usuaris**.

Per què necessitem la concurrència?

Com heu vist el món real actua com un gran sistema paral·lel. Per tant, tots els sistemes de temps real són inherentment concurrents i lògicament els nostres sistemes informàtics també han de poder actuar en paral·lel. En la figura, podeu observar com una tasca es descomposa en diferents parts per poder aprofitar els diferents processadors i nuclis de cada processador.

L'execució **concurrent** recull un conjunt de tècniques informàtiques usades per representar i gestionar el paral·lisme i les eines de sincronització i comunicació entre programes.

Què és un sistema informàtic?

Un **sistema informàtic** és la interconnexió d'elements de maquinari per exemple **d'1 o més CPU, memòria** i components E/S ... Amb la finalitat d'executar programes i accions (en sèrie o de forma concurrent) per **1 o múltiples usuaris**.

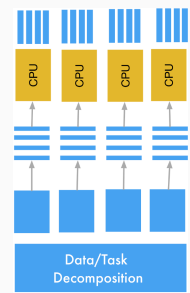


Figura 9: Esquema de sistemes paral·lels

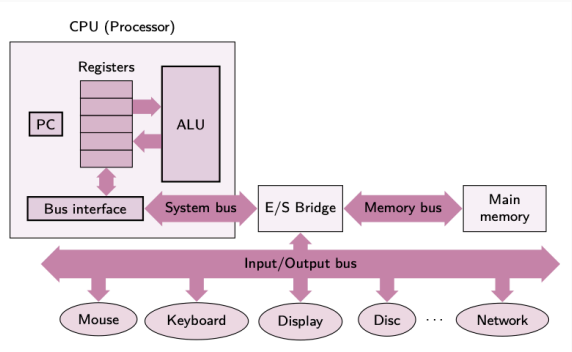


Figura 10: Esquema d'un sistema informàtic

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└ Introducció a la temàtica

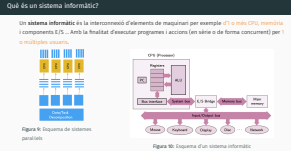
└ Què és un sistema informàtic?

Un **sistema informàtic** és la interconnexió d'elements de maquinari per exemple **d'1 o més CPU, memòria** i components E/S ... Amb la finalitat d'executar programes i accions (en sèrie o de forma concurrent) per **1 o múltiples usuaris**.

Per què necessitem la concurrència?

Com heu vist el món real actua com un gran sistema paral·lel. Per tant, tots els sistemes de temps real són inherentment concurrents i lògicament els nostres sistemes informàtics també han de poder actuar en paral·lel. En la figura, podeu observar com una tasca es descomposa en diferents parts per poder aprofitar els diferents processadors i nuclis de cada processador.

L'execució **concurrent** recull un conjunt de tècniques informàtiques usades per representar i gestionar el paral·lisme i les eines de sincronització i comunicació entre programes.



Quantes vegades com a usuaris d'un sistema informàtic llegiu un fitxer?

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Gestió d'un sistema informàtic (I)

Quantes vegades com a usuari d'un sistema informàtic llegiu un fitxer?

Imageneu que sou un sistema informàtic i us envien l'ordre de llegir un fitxer? Quins és el procediment per fer aquesta acció?

Quantes vegades com a usuari d'un sistema informàtic llegiu un fitxer?

Imageneu que sou un sistema informàtic i us envien l'ordre de llegir un fitxer? Quins és el procediment per fer aquesta acció?

Quantes vegades com a usuaris d'un sistema informàtic llegiu un fitxer?

Moltes vegades...!

Cada usuari de forma activa (llegir un document) o passiva (el vostre sistema operatiu llegeix molts fitxers únicament per arrancar...).

Per fer-ho el sistema opartiu té que indicar al disc que engegi el seu motor, es situi a la posició a llegir, llegeixi aquella posició i la transfereixi, i finalment apagui el motor.

Té sentit que un usuari cada cop que llegeix un fitxer hagi de tenir en compte aquest procediment i realitzar-lo de forma rutinària? Necessitem un programari que ens permeti efectuar accions rutinàries de forma automàtica i transparent a l'usuari. Si no els sistemes informàtics no serien usables.

Quantes vegades com a usuaris d'un sistema informàtic llegiu un fitxer?

Moltes vegades...!

Imagineu que sou un sistema informàtic i us envien l'ordre de llegir un fitxer? Quins és el procediment per fer aquesta acció?

Cada usuari de forma activa (llegir un document) o passiva (el vostre sistema operatiu llegeix molts fitxers únicament per arrancar...).

Per fer-ho el sistema opartiu té que indicar al disc que engegi el seu motor, es situi a la posició a llegir, llegeixi aquella posició i la transfereixi, i finalment apagui el motor.

Té sentit que un usuari cada cop que llegeix un fitxer hagi de tenir en compte aquest procediment i realitzar-lo de forma rutinària? Necessitem un programari que ens permeti efectuar accions rutinàries de forma automàtica i transparent a l'usuari. Si no els sistemes informàtics no serien usables.

Quantes vegades com a usuaris d'un sistema informàtic llegiu un fitxer?

Moltes vegades...!

Imagineu que sou un sistema informàtic i us envien l'ordre de llegir un fitxer? Quins és el procediment per fer aquesta acció?

- Engegar motor del disc.

Cada usuari de forma activa (llegir un document) o passiva (el vostre sistema operatiu llegeix molts fitxers únicament per arrancar...).

Per fer-ho el sistema opartiu té que indicar al disc que engegi el seu motor, es situi a la posició a llegir, llegeixi aquella posició i la transfereixi, i finalment apagui el motor.

Té sentit que un usuari cada cop que llegeix un fitxer hagi de tenir en compte aquest procediment i realitzar-lo de forma rutinària? Necessitem un programari que ens permeti efectuar accions rutinàries de forma automàtica i transparent a l'usuari. Si no els sistemes informàtics no serien usables.

Gestió d'un sistema informàtic (II)

Quantes vegades com a usuaris d'un sistema informàtic llegiu un fitxer?

Moltes vegades...!

Imagineu que sou un sistema informàtic i us envien l'ordre de llegir un fitxer? Quins és el procediment per fer aquesta acció?

- Engegar motor del disc.

Quantes vegades com a usuaris d'un sistema informàtic llegiu un fitxer?

Moltes vegades...!

Imagineu que sou un sistema informàtic i us envien l'ordre de llegir un fitxer? Quins és el procediment per fer aquesta acció?

- Engegar motor del disc.
- Buscar posició al disc a llegir (pista, cara, sector).

Cada usuari de forma activa (llegir un document) o passiva (el vostre sistema operatiu llegeix molts fitxers únicament per arrancar...).

Per fer-ho el sistema opartiu té que indicar al disc que engegi el seu motor, es situi a la posició a llegir, llegeixi aquella posició i la transfereixi, i finalment apagui el motor.

Té sentit que un usuari cada cop que llegeix un fitxer hagi de tenir en compte aquest procediment i realitzar-lo de forma rutinària? Necessitem un programari que ens permeti efectuar accions rutinàries de forma automàtica i transparent a l'usuari. Si no els sistemes informàtics no serien usables.

Gestió d'un sistema informàtic (II)

Quantes vegades com a usuaris d'un sistema informàtic llegiu un fitxer?

Moltes vegades...!

Imagineu que sou un sistema informàtic i us envien l'ordre de llegir un fitxer? Quins és el procediment per fer aquesta acció?

- Engegar motor del disc.
- Buscar posició al disc a llegir (pista, cara, sector).

Quantes vegades com a usuaris d'un sistema informàtic llegiu un fitxer?

Moltes vegades...!

Imagineu que sou un sistema informàtic i us envien l'ordre de llegir un fitxer? Quins és el procediment per fer aquesta acció?

- Engegar motor del disc.
- Buscar posició al disc a llegir (pista, cara, sector).
- Llegir.

Cada usuari de forma activa (llegir un document) o passiva (el vostre sistema operatiu llegeix molts fitxers únicament per arrancar...).

Per fer-ho el sistema opartiu té que indicar al disc que engegi el seu motor, es situi a la posició a llegir, llegeixi aquella posició i la transfereixi, i finalment apagui el motor.

Té sentit que un usuari cada cop que llegeix un fitxer hagi de tenir en compte aquest procediment i realitzar-lo de forma rutinària? Necessitem un programari que ens permeti efectuar accions rutinàries de forma automàtica i transparent a l'usuari. Si no els sistemes informàtics no serien usables.

2024-09-30

Gestió d'un sistema informàtic (II)

Quantes vegades com a usuaris d'un sistema informàtic llegiu un fitxer?

Moltes vegades...!

Imagineu que sou un sistema informàtic i us envien l'ordre de llegir un fitxer? Quins és el procediment per fer aquesta acció?

- Engegar motor del disc.
- Buscar posició al disc a llegir (pista, cara, sector).
- Llegir.

Quantes vegades com a usuaris d'un sistema informàtic llegiu un fitxer?

Moltes vegades...!

Imagineu que sou un sistema informàtic i us envien l'ordre de llegir un fitxer? Quins és el procediment per fer aquesta acció?

- Engegar motor del disc.
- Buscar posició al disc a llegir (pista, cara, sector).
- Llegir.
- Apagar el motor.

Cada usuari de forma activa (llegir un document) o passiva (el vostre sistema operatiu llegeix molts fitxers únicament per arrancar...).

Per fer-ho el sistema opartiu té que indicar al disc que engegi el seu motor, es situi a la posició a llegir, llegeixi aquella posició i la transfereixi, i finalment apagui el motor.

Té sentit que un usuari cada cop que llegeix un fitxer hagi de tenir en compte aquest procediment i realitzar-lo de forma rutinària? Necessitem un programari que ens permeti efectuar accions rutinàries de forma automàtica i transparent a l'usuari. Si no els sistemes informàtics no serien usables.

2024-09-30

Gestió d'un sistema informàtic (II)

Quantes vegades com a usuaris d'un sistema informàtic llegiu un fitxer?

Moltes vegades...!

Imagineu que sou un sistema informàtic i us envien l'ordre de llegir un fitxer? Quins és el procediment per fer aquesta acció?

- Engegar motor del disc.
- Buscar posició al disc a llegir (pista, cara, sector).
- Llegir.
- Apagar el motor.

Cada peça de hardware és diferent ➡ La complexitat per gestionar els recursos és molt elevada.

Us imagineu haver d’escriure software per cada permutació d’elements diferents del sistema informàtic? Això seria un desastre... Quantes permutacions de diferents components podem tenir avui en dia? OMG! Necessitem alguna cosa que ens permeti superar aquest obstacle de forma intel·ligent i ens permeti desenvolupar software de forma independent al hardware.

Cada peça de hardware és diferent ➡ La complexitat per gestionar els recursos és molt elevada.

- Arquitectures diferents de processadors i també de generacions (x86,ARM, RISC-V,MIPS, PowerPC,...).

Us imagineu haver d’escriure software per cada permutació d’elements diferents del sistema informàtic? Això seria un desastre... Quantes permutacions de diferents components podem tenir avui en dia? OMG! Necessitem alguna cosa que ens permeti superar aquest obstacle de forma intel·ligent i ens permeti desenvolupar software de forma independent al hardware.

Cada peça de hardware és diferent ➡ La complexitat per gestionar els recursos és molt elevada.

- Arquitectures diferents de processadors i també de generacions (x86,ARM, RISC-V,MIPS, PowerPC,...).
- Diferents tipus de memòries (RAM DDR3, DDR4, DDR5, NAND,...).

Us imagineu haver d’escriure software per cada permutació d’elements diferents del sistema informàtic? Això seria un desastre... Quantes permutacions de diferents components podem tenir avui en dia? OMG! Necessitem alguna cosa que ens permeti superar aquest obstacle de forma intel·ligent i ens permeti desenvolupar software de forma independent al hardware.

Cada peça de hardware és diferent ➡ La complexitat per gestionar els recursos és molt elevada.

- Arquitectures diferents de processadors i també de generacions (x86,ARM, RISC-V,MIPS, PowerPC,...).
- Diferents tipus de memòries (RAM DDR3, DDR4, DDR5, NAND,...).

Cada peça de hardware és diferent ➡ La complexitat per gestionar els recursos és molt elevada.

- Arquitectures diferents de processadors i també de generacions (x86,ARM, RISC-V,MIPS, PowerPC,...).
- Diferents tipus de memòries (RAM DDR3, DDR4, DDR5, NAND,...).
- Diferents tipus de discs (HDD o SSD).

Us imagineu haver d’escriure software per cada permutació d’elements diferents del sistema informàtic? Això seria un desastre... Quantes permutacions de diferents components podem tenir avui en dia? OMG! Necessitem alguna cosa que ens permeti superar aquest obstacle de forma intel·ligent i ens permeti desenvolupar software de forma independent al hardware.

Cada peça de hardware és diferent ➡ La complexitat per gestionar els recursos és molt elevada.

- Arquitectures diferents de processadors i també de generacions (x86,ARM, RISC-V,MIPS, PowerPC,...).
- Diferents tipus de memòries (RAM DDR3, DDR4, DDR5, NAND,...).
- Diferents tipus de discs (HDD o SSD).
- Diferents dispositius entrada/sortida.

Us imagineu haver d’escriure software per cada permutació d’elements diferents del sistema informàtic? Això seria un desastre... Quantes permutacions de diferents components podem tenir avui en dia? OMG! Necessitem alguna cosa que ens permeti superar aquest obstacle de forma intel·ligent i ens permeti desenvolupar software de forma independent al hardware.

Complexitat dels sistemes informàtics (I)

Cada peça de hardware és diferent ➡ La complexitat per gestionar els recursos és molt elevada.

- Arquitectures diferents de processadors i també de generacions (x86,ARM, RISC-V,MIPS, PowerPC,...).
- Diferents tipus de memòries (RAM DDR3, DDR4, DDR5, NAND,...).
- Diferents tipus de discs (HDD o SSD).
- Diferents dispositius entrada/sortida.

Cada peça de hardware és diferent ➡ La complexitat per gestionar els recursos és molt elevada.

- Arquitectures diferents de processadors i també de generacions (x86,ARM, RISC-V,MIPS, PowerPC,...).
- Diferents tipus de memòries (RAM DDR3, DDR4, DDR5, NAND,...).
- Diferents tipus de discs (HDD o SSD).
- Diferents dispositius entrada/sortida.
- Diferents entorns de xarxa.

Us imagineu haver d’escriure software per cada permutació d’elements diferents del sistema informàtic? Això seria un desastre... Quantes permutacions de diferents components podem tenir avui en dia? OMG! Necessitem alguna cosa que ens permeti superar aquest obstacle de forma intel·ligent i ens permeti desenvolupar software de forma independent al hardware.

Complexitat dels sistemes informàtics (I)

Cada peça de hardware és diferent ➡ La complexitat per gestionar els recursos és molt elevada.

- Arquitectures diferents de processadors i també de generacions (x86,ARM, RISC-V,MIPS, PowerPC,...).
- Diferents tipus de memòries (RAM DDR3, DDR4, DDR5, NAND,...).
- Diferents tipus de discs (HDD o SSD).
- Diferents dispositius entrada/sortida.
- Diferents entorns de xarxa.

Cada peça de hardware és diferent ➡ La complexitat per gestionar els recursos és molt elevada.

- Arquitectures diferents de processadors i també de generacions (x86,ARM, RISC-V,MIPS, PowerPC,...).
- Diferents tipus de memòries (RAM DDR3, DDR4, DDR5, NAND,...).
- Diferents tipus de discs (HDD o SSD).
- Diferents dispositius entrada/sortida.
- Diferents entorns de xarxa.

... entre moltes altres ...

Us imagineu haver d’escriure software per cada permutació d’elements diferents del sistema informàtic? Això seria un desastre... Quantes permutacions de diferents components podem tenir avui en dia? OMG! Necessitem alguna cosa que ens permeti superar aquest obstacle de forma intel·ligent i ens permeti desenvolupar software de forma independent al hardware.

Complexitat dels sistemes informàtics (I)

Cada peça de hardware és diferent ➡ La complexitat per gestionar els recursos és molt elevada.

- Arquitectures diferents de processadors i també de generacions (x86,ARM, RISC-V,MIPS, PowerPC,...).
- Diferents tipus de memòries (RAM DDR3, DDR4, DDR5, NAND,...).
- Diferents tipus de discs (HDD o SSD).
- Diferents dispositius entrada/sortida.
- Diferents entorns de xarxa.

... entre moltes altres ...

Tots els programes necessiten accedir a tot el hardware?

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Complexitat dels sistemes informàtics (II)

Tots els programes necessiten accedir a tot el hardware?

Què pot passar si un programa pot accedir a tota la RAM?

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Complexitat dels sistemes informàtics (II)

Tots els programes necessiten accedir a tot el hardware?

Què pot passar si un programa pot accedir a tota la RAM?

Tots els programes necessiten accedir a tot el hardware?

No, clar que no!

Seguretat:

- Si un programa controla tot el sistema, els altres no poden accedir-hi (Denegació de serveis).
- Diferents usuaris poden tenir diferents dades; si un programa pot accedir a tot, podria veure informació restringida.
- No necessitem accés a tot el hardware per realitzar les nostres tasques.
- Un usuari malintencionat podria danyar el sistema amb un procés defectuós.

Tots els programes necessiten accedir a tot el hardware?

Per suposat que no! Això podria causar problemes de seguretat com:

No, clar que no!

Seguretat:

- Si un programa controla tot el sistema, els altres no poden accedir-hi (Denegació de serveis).
- Diferents usuaris poden tenir diferents dades; si un programa pot accedir a tot, podria veure informació restringida.
- No necessitem accés a tot el hardware per realitzar les nostres tasques.
- Un usuari malintencionat podria danyar el sistema amb un procés defectuós.

Tots els programes necessiten accedir a tot el hardware?

Per suposat que no! Això podria causar problemes de seguretat com:

- **Pèrdua de confidencialitat:** Accés no autoritzat a dades sensibles.

No, clar que no!

Seguretat:

- Si un programa controla tot el sistema, els altres no poden accedir-hi (Denegació de serveis).
- Diferents usuaris poden tenir diferents dades; si un programa pot accedir a tot, podria veure informació restringida.
- No necessitem accés a tot el hardware per realitzar les nostres tasques.
- Un usuari malintencionat podria danyar el sistema amb un procés defectuós.

Tots els programes necessiten accedir a tot el hardware?

Per suposat que no! Això podria causar problemes de seguretat com:

- **Pèrdua de confidencialitat:** Accés no autoritzat a dades sensibles.
- **Accés a informació restringida:** Usuaris no autoritzats podrien veure dades privades.

No, clar que no!

Seguretat:

- Si un programa controla tot el sistema, els altres no poden accedir-hi (Denegació de serveis).
- Diferents usuaris poden tenir diferents dades; si un programa pot accedir a tot, podria veure informació restringida.
- No necessitem accés a tot el hardware per realitzar les nostres tasques.
- Un usuari malintencionat podria danyar el sistema amb un procés defectuós.

Tots els programes necessiten accedir a tot el hardware?

Per suposat que no! Això podria causar problemes de seguretat com:

- **Pèrdua de confidencialitat:** Accés no autoritzat a dades sensibles.
- **Accés a informació restringida:** Usuaris no autoritzats podrien veure dades privades.
- **Denegació de serveis:** Un programa podria bloquejar l'accés a recursos per a altres programes.

No, clar que no!

Seguretat:

- Si un programa controla tot el sistema, els altres no poden accedir-hi (Denegació de serveis).
- Diferents usuaris poden tenir diferents dades; si un programa pot accedir a tot, podria veure informació restringida.
- No necessitem accés a tot el hardware per realitzar les nostres tasques.
- Un usuari malintencionat podria danyar el sistema amb un procés defectuós.

Tots els programes necessiten accedir a tot el hardware?

Per suposat que no! Això podria causar problemes de seguretat com:

- **Pèrdua de confidencialitat:** Accés no autoritzat a dades sensibles.
- **Accés a informació restringida:** Usuaris no autoritzats podrien veure dades privades.
- **Denegació de serveis:** Un programa podria bloquejar l'accés a recursos per a altres programes.

Tots els programes necessiten accedir a tot el hardware?

Per suposat que no! Això podria causar problemes de seguretat com:

- **Pèrdua de confidencialitat:** Accés no autoritzat a dades sensibles.
- **Accés a informació restringida:** Usuaris no autoritzats podrien veure dades privades.
- **Denegació de serveis:** Un programa podria bloquejar l'accés a recursos per a altres programes.

Què pot passar si un programa pot accedir a tota la RAM?

No, clar que no!

Seguretat:

- Si un programa controla tot el sistema, els altres no poden accedir-hi (Denegació de serveis).
- Diferents usuaris poden tenir diferents dades; si un programa pot accedir a tot, podria veure informació restringida.
- No necessitem accés a tot el hardware per realitzar les nostres tasques.
- Un usuari malintencionat podria danyar el sistema amb un procés defectuós.

2024-09-30

Tots els programes necessiten accedir a tot el hardware?

Per suposat que no! Això podria causar problemes de seguretat com:

- **Pèrdua de confidencialitat:** Accés no autoritzat a dades sensibles.
- **Accés a informació restringida:** Usuaris no autoritzats podrien veure dades privades.
- **Denegació de serveis:** Un programa podria bloquejar l'accés a recursos per a altres programes.

Què pot passar si un programa pot accedir a tota la RAM?

Tots els programes necessiten accedir a tot el hardware?

Per suposat que no! Això podria causar problemes de seguretat com:

- **Pèrdua de confidencialitat:** Accés no autoritzat a dades sensibles.
- **Accés a informació restringida:** Usuaris no autoritzats podrien veure dades privades.
- **Denegació de serveis:** Un programa podria bloquejar l'accés a recursos per a altres programes.

Què pot passar si un programa pot accedir a tota la RAM?

- Un programa de l'usuari Jordi amb accés a tota la RAM podria veure les dades del programa de l'usuari Pere.

No, clar que no!

Seguretat:

- Si un programa controla tot el sistema, els altres no poden accedir-hi (Denegació de serveis).
- Diferents usuaris poden tenir diferents dades; si un programa pot accedir a tot, podria veure informació restringida.
- No necessitem accés a tot el hardware per realitzar les nostres tasques.
- Un usuari malintencionat podria danyar el sistema amb un procés defectuós.

Complexitat dels sistemes informàtics (III)

Tots els programes necessiten accedir a tot el hardware?

Per suposat que no! Això podria causar problemes de seguretat com:

- **Pèrdua de confidencialitat:** Accés no autoritzat a dades sensibles.
- **Accés a informació restringida:** Usuaris no autoritzats podrien veure dades privades.
- **Denegació de serveis:** Un programa podria bloquejar l'accés a recursos per a altres programes.

Què pot passar si un programa pot accedir a tota la RAM?

- Un programa de l'usuari Jordi amb accés a tota la RAM podria veure les dades del programa de l'usuari Pere.

2024-09-30

Tots els programes necessiten accedir a tot el hardware?

Per suposat que no! Això podria causar problemes de seguretat com:

- **Pèrdua de confidencialitat:** Accés no autoritzat a dades sensibles.
- **Accés a informació restringida:** Usuaris no autoritzats podrien veure dades privades.
- **Denegació de serveis:** Un programa podria bloquejar l'accés a recursos per a altres programes.

Què pot passar si un programa pot accedir a tota la RAM?

- Un programa de l'usuari Jordi amb accés a tota la RAM podria veure les dades del programa de l'usuari Pere.
- Si el programa falla, podria afectar tot el sistema i requerir un reinici.

No, clar que no!

Seguretat:

- Si un programa controla tot el sistema, els altres no poden accedir-hi (Denegació de serveis).
- Diferents usuaris poden tenir diferents dades; si un programa pot accedir a tot, podria veure informació restringida.
- No necessitem accés a tot el hardware per realitzar les nostres tasques.
- Un usuari malintencionat podria danyar el sistema amb un procés defectuós.

2024-09-30

Complexitat dels sistemes informàtics (III)

Tots els programes necessiten accedir a tot el hardware?

Per suposat que no! Això podria causar problemes de seguretat com:

- **Pèrdua de confidencialitat:** Accés no autoritzat a dades sensibles.
- **Accés a informació restringida:** Usuaris no autoritzats podrien veure dades privades.
- **Denegació de serveis:** Un programa podria bloquejar l'accés a recursos per a altres programes.

Què pot passar si un programa pot accedir a tota la RAM?

- Un programa de l'usuari Jordi amb accés a tota la RAM podria veure les dades del programa de l'usuari Pere.
- Si el programa falla, podria afectar tot el sistema i requerir un reinici.

Un programa pot fer fallar tot el sistema?

Un programa o procés pot fer fallar tot el sistema si no està ben dissenyat. Per tant, és important tenir en compte aqueixes situacions per evitar-les.

En el primer exemple, tenim un bucle infinit. Aquesta situació en els sistemes linux actuals no és un problema, ja que el sistema operatiu pot gestionar aquest tipus de situacions. No obstant això, en sistemes més antics, aquest tipus de bucles podrien fer que el sistema no respongués.

En el segon exemple, tenim un bucle infinit i una crida a la funció fork(). Aquesta funció crea un nou procés que és una còpia exacta del procés pare. Això podria fer que el sistema es saturés amb molts processos i no pogués respondre. Aquesta situació podria requerir un reinici del sistema per recuperar-lo i es un problema en els sistemes actuals.

Fixeu-vos com un simple programa pot fer fallar tot el sistema.

Un programa pot fer fallar tot el sistema?

```
int main(){
    while(1);
}
```

Un programa o procés pot fer fallar tot el sistema si no està ben dissenyat. Per tant, és important tenir en compte aqueixes situacions per evitar-les.

En el primer exemple, tenim un bucle infinit. Aquesta situació en els sistemes linux actuals no és un problema, ja que el sistema operatiu pot gestionar aquest tipus de situacions. No obstant això, en sistemes més antics, aquest tipus de bucles podrien fer que el sistema no respongués.

En el segon exemple, tenim un bucle infinit i una crida a la funció fork(). Aquesta funció crea un nou procés que és una còpia exacta del procés pare. Això podria fer que el sistema es saturés amb molts processos i no pogués respondre. Aquesta situació podria requerir un reinici del sistema per recuperar-lo i es un problema en els sistemes actuals.

Fixeu-vos com un simple programa pot fer fallar tot el sistema.

```
int main(){
    while(1);
}
```

Un programa pot fer fallar tot el sistema?

```
int main(){
    while(1);
}
```

```
int main(){
    while(1);
    fork();
}
```

2024-09-30

Un programa o procés pot fer fallar tot el sistema si no està ben dissenyat. Per tant, és important tenir en compte aqueixes situacions per evitar-les.

En el primer exemple, tenim un bucle infinit. Aquesta situació en els sistemes linux actuals no és un problema, ja que el sistema operatiu pot gestionar aquest tipus de situacions. No obstant això, en sistemes més antics, aquest tipus de bucles podrien fer que el sistema no respongués.

En el segon exemple, tenim un bucle infinit i una crida a la funció fork(). Aquesta funció crea un nou procés que és una còpia exacta del procés pare. Això podria fer que el sistema es saturés amb molts processos i no pogués respondre. Aquesta situació podria requerir un reinici del sistema per recuperar-lo i es un problema en els sistemes actuals.

Fixeu-vos com un simple programa pot fer fallar tot el sistema.

Un programa pot fer fallar tot el sistema?

```
int main(){
    while(1);
}
```

```
int main(){
    while(1);
    fork();
}
```

Què és un sistema operatiu?

Un **sistema operatiu (SO)** és una capa de software que permet la comunicació i la gestió del maquinari habilitant als usuaris l’execució de programes. El SO actua d’**intermediari (interfície)** entre els **usuaris i el maquinari**.

Unitat 0 · Sistemes Operatius (SO)

2024-09-30

└─ Introducció a la temàtica

└─ Què és un sistema operatiu?

Un **sistema operatiu (SO)** és una capa de software que permet la comunicació i la gestió del maquinari habilitant als usuaris l’execució de programes. El SO actua d’**intermediari (interfície)** entre els **usuaris i el maquinari**. Per tant, és un programa que controla el maquinari i permet als usuaris interactuar amb ell. Aquesta interacció pot ser a través d’una interfície gràfica d’usuari (GUI) o una interfície de línia de comandes (CLI). També, gestiona els recursos del sistema, com la memòria, el disc dur i els dispositius d’entrada/sortida. A més, el SO proporciona funcionalitats com la gestió de fitxers, la seguretat i la xarxa.

Podem dir que el SO és el cervell del sistema informàtic, ja que controla tots els components i permet als usuaris interactuar amb ells. Sense un SO, els usuaris no podrien utilitzar el maquinari del sistema de manera eficient i segura.

Què és un sistema operatiu?
Un **sistema operatiu (SO)** és una capa de software que permet la comunicació i la gestió del maquinari habilitant als usuaris l’execució de programes. El SO actua d’**intermediari (interfície)** entre els **usuaris i el maquinari**.

Què és un sistema operatiu?

Un **sistema operatiu (SO)** és una capa de software que permet la comunicació i la gestió del maquinari habilitant als usuaris l'execució de programes. El SO actua d'**intermediari (interfície)** entre els **usuaris** i el **maquinari**.

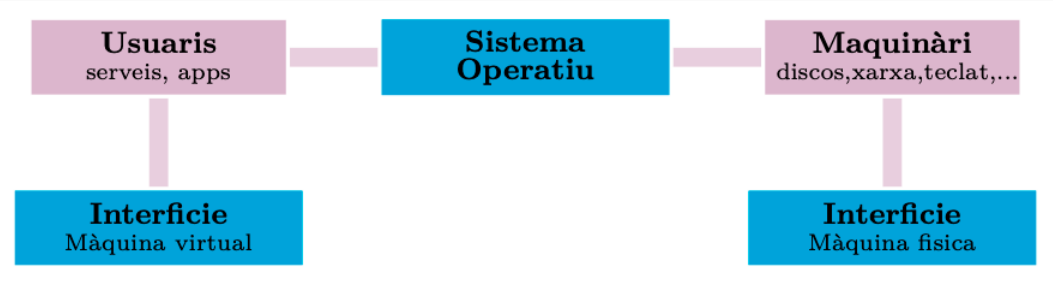
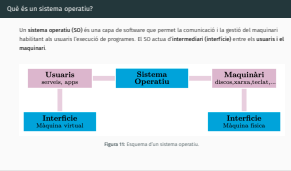


Figura 11: Esquema d'un sistema operatiu.

- └ Introducció a la temàtica
- └ Què és un sistema operatiu?

2024-09-30



Un **sistema operatiu (SO)** és una capa de software que permet la comunicació i la gestió del maquinari habilitant als usuaris l'execució de programes. El SO actua d'**intermediari (interfície)** entre els **usuaris** i el **maquinari**. Per tant, és un programa que controla el maquinari i permet als usuaris interactuar amb ell. Aquesta interacció pot ser a través d'una interfície gràfica d'usuari (GUI) o una interfície de línia de comandes (CLI). També, gestiona els recursos del sistema, com la memòria, el disc dur i els dispositius d'entrada/sortida. A més, el SO proporciona funcionalitats com la gestió de fitxers, la seguretat i la xarxa. Podem dir que el SO és el cervell del sistema informàtic, ja que controla tots els components i permet als usuaris interactuar amb ells. Sense un SO, els usuaris no podrien utilitzar el maquinari del sistema de manera eficient i segura.

Què volen/necessiten els programadors?

Una plataforma

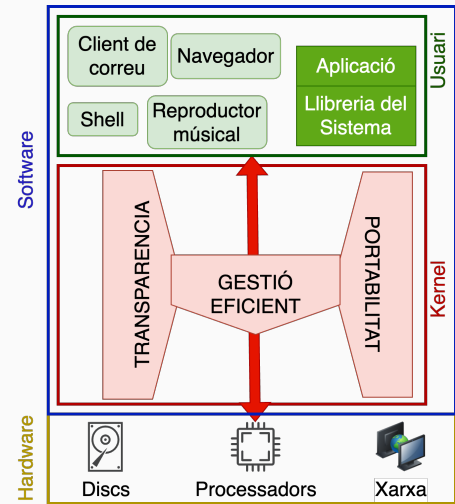


Figura 12: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què volen/necessiten els programadors?

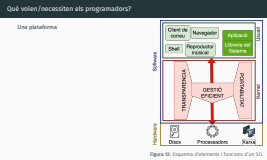


Figura 12: Esquema d'elements i funcions d'un SO.

Què volen/necessiten els programadors?

Una plataforma

- Una plataforma per executar aplicacions.

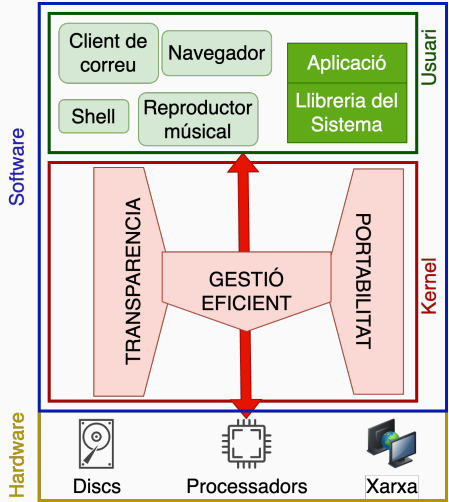


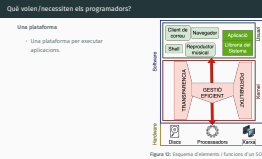
Figura 12: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què volen/necessiten els programadors?



Què volen/necessiten els programadors?

Una plataforma

- Una plataforma per executar aplicacions.
- Una plataforma transparent per evitar la complexitat del hardware.

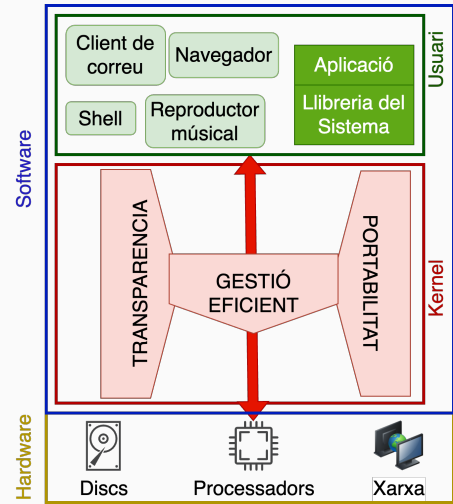


Figura 12: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què volen/necessiten els programadors?

Què volen/necessiten els programadors?

Una plataforma

- Una plataforma per executar aplicacions.
- Una plataforma transparent per evitar la complexitat del hardware.

Figura 12: Esquema d'elements i funcions d'un SO.

Què volen/necessiten els programadors?

Una plataforma

- Una plataforma per executar aplicacions.
- Una plataforma transparent per evitar la complexitat del hardware.
- Una plataforma eficient per utilitzar els recursos de forma òptima.

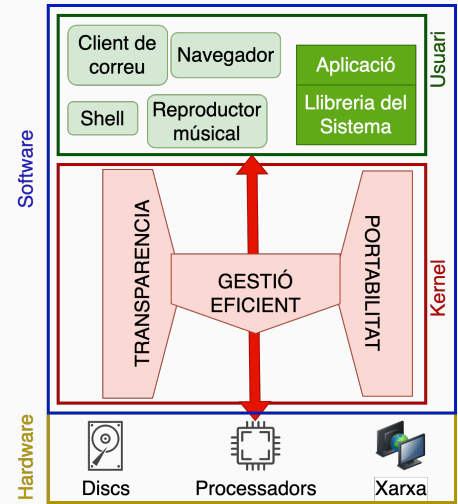


Figura 12: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què volen/necessiten els programadors?

Què volen/necessiten els programadors?

Una plataforma

- Una plataforma per executar aplicacions.
- Una plataforma transparent per evitar la complexitat del hardware.
- Una plataforma eficient per utilitzar els recursos de forma òptima.

Figura 12: Esquema d'elements i funcions d'un SO.

Què volen/necessiten els programadors?

Una plataforma

- Una plataforma per executar aplicacions.
- Una plataforma transparent per evitar la complexitat del hardware.
- Una plataforma eficient per utilitzar els recursos de forma òptima.
- Una plataforma portable per utilitzar-ho independentment del hardware.

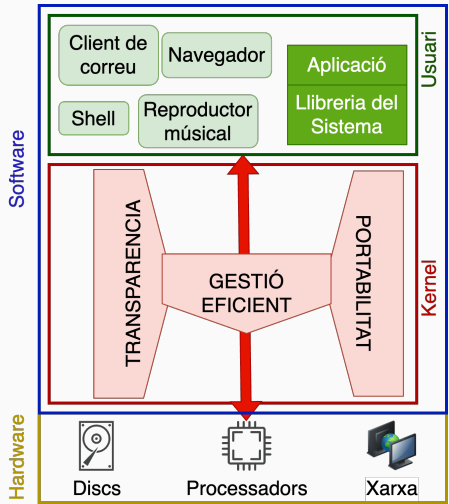


Figura 12: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què volen/necessiten els programadors?

Què volen/necessiten els programadors?

- Una plataforma per executar aplicacions.
- Una plataforma transparent per evitar la complexitat del hardware.
- Una plataforma eficient per utilitzar els recursos de forma òptima.
- Una plataforma portable per utilitzar-ho independentment del hardware.

Figura 12: Esquema d'elements i funcions d'un SO.

Què ofereix el sistema operatiu?

Serveis

- Controlar usuaris i aplicacions.

Garanties

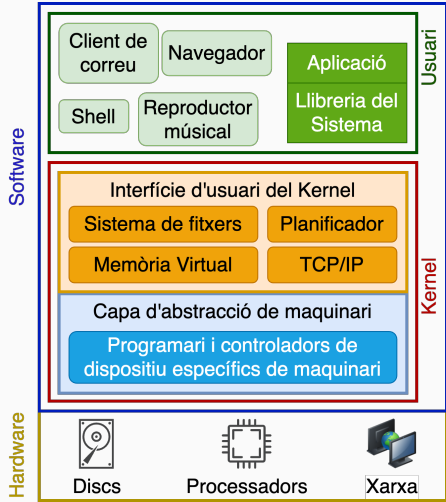


Figura 13: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què ofereix el sistema operatiu?

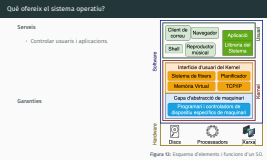


Figura 13: Esquema d'elements i funcions d'un SO.

Què ofereix el sistema operatiu?

Serveis

- Controlar usuaris i aplicacions.
- Gestionar usuaris i aplicacions.

Garanties

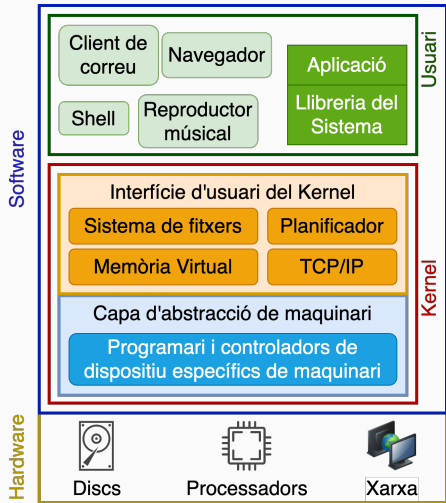


Figura 13: Esquema d'elements i funcions d'un SO.

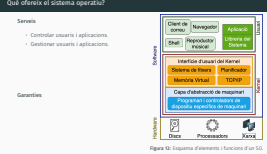


Figura 13: Esquema d'elements i funcions d'un SO.

Què ofereix el sistema operatiu?

Serveis

- Controlar usuaris i aplicacions.
- Gestionar usuaris i aplicacions.
- Gestionar la Memòria.

Garanties

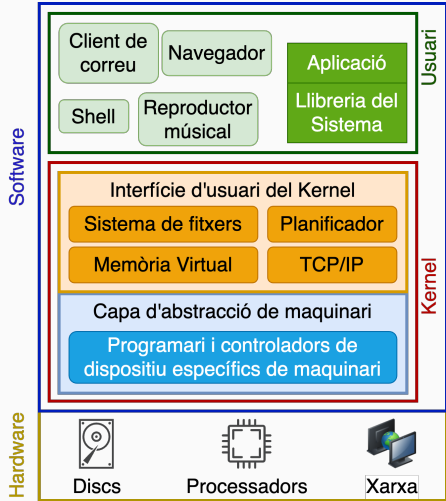


Figura 13: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què ofereix el sistema operatiu?

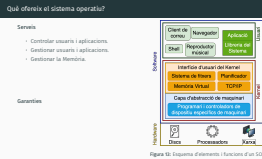


Figura 13: Esquema d'elements i funcions d'un SO.

Què ofereix el sistema operatiu?

Serveis

- Controlar usuaris i aplicacions.
- Gestionar usuaris i aplicacions.
- Gestionar la Memòria.
- Sistema de fitxers.

Garanties

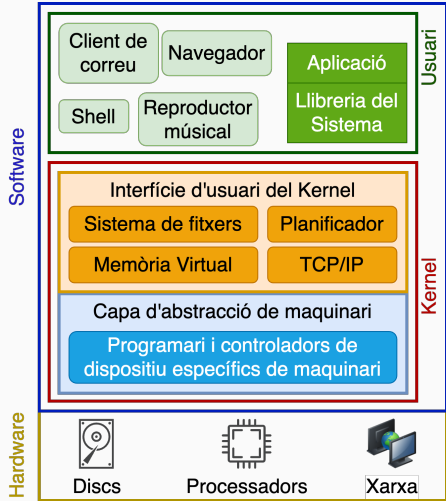


Figura 13: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què ofereix el sistema operatiu?

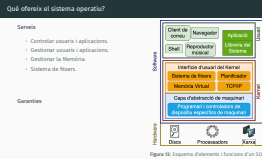


Figura 13: Esquema d'elements i funcions d'un SO.

Què ofereix el sistema operatiu?

Serveis

- Controlar usuaris i aplicacions.
- Gestionar usuaris i aplicacions.
- Gestionar la Memòria.
- Sistema de fitxers.
- Planificadors.

Garanties

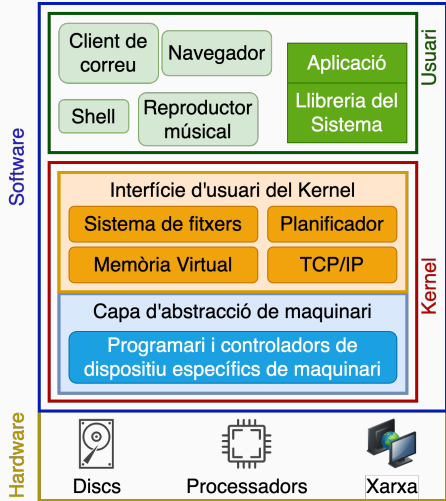


Figura 13: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què ofereix el sistema operatiu?

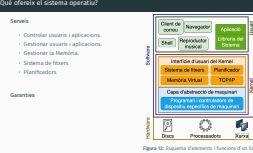


Figura 13: Esquema d'elements i funcions d'un SO.

Què ofereix el sistema operatiu?

Serveis

- Controlar usuaris i aplicacions.
- Gestionar usuaris i aplicacions.
- Gestionar la Memòria.
- Sistema de fitxers.
- Planificadors.
- Eines de xarxa.

Garanties

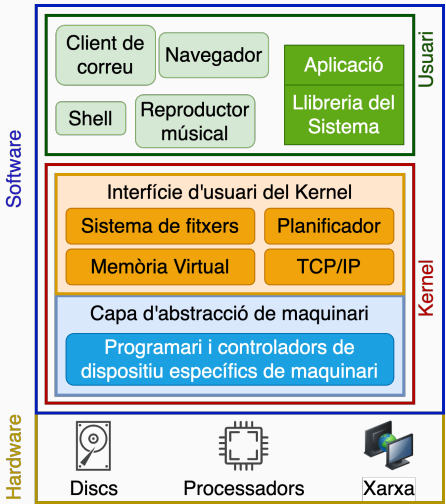


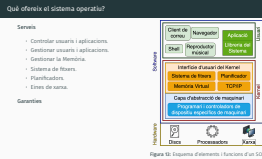
Figura 13: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què ofereix el sistema operatiu?



Què ofereix el sistema operatiu?

Serveis

- Controlar usuaris i aplicacions.
- Gestionar usuaris i aplicacions.
- Gestionar la Memòria.
- Sistema de fitxers.
- Planificadors.
- Eines de xarxa.

Garanties

- Seguretat.

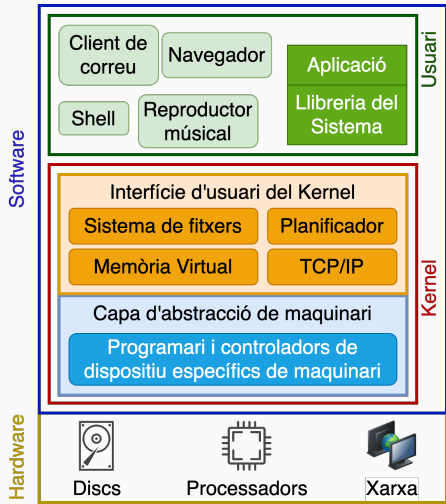


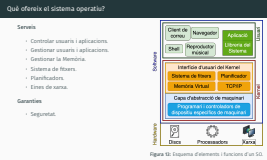
Figura 13: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què ofereix el sistema operatiu?



Què ofereix el sistema operatiu?

Serveis

- Controlar usuaris i aplicacions.
- Gestionar usuaris i aplicacions.
- Gestionar la Memòria.
- Sistema de fitxers.
- Planificadors.
- Eines de xarxa.

Garanties

- Seguretat.
- Transparència.

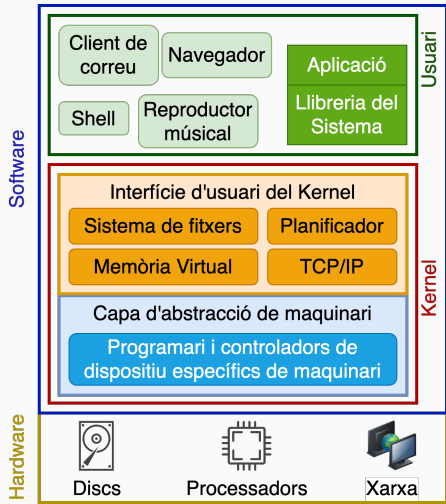


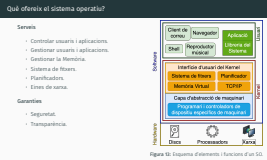
Figura 13: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què ofereix el sistema operatiu?



Què ofereix el sistema operatiu?

Serveis

- Controlar usuaris i aplicacions.
- Gestionar usuaris i aplicacions.
- Gestionar la Memòria.
- Sistema de fitxers.
- Planificadors.
- Eines de xarxa.

Garanties

- Seguretat.
- Transparència.
- Eficiència.

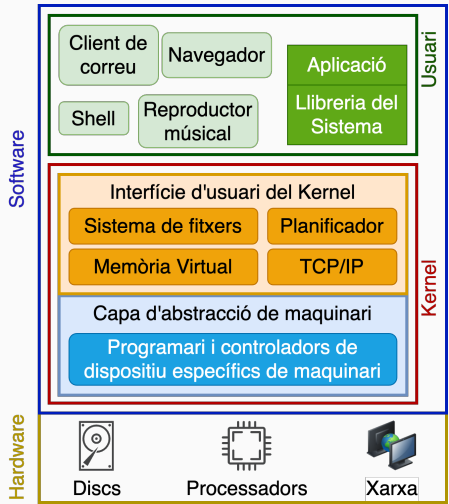


Figura 13: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què ofereix el sistema operatiu?

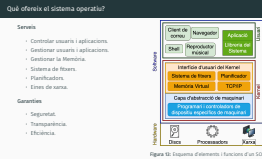


Figura 14: Esquema d'elements i funcions d'un SO.

Què ofereix el sistema operatiu?

Serveis

- Controlar usuaris i aplicacions.
- Gestionar usuaris i aplicacions.
- Gestionar la Memòria.
- Sistema de fitxers.
- Planificadors.
- Eines de xarxa.

Garanties

- Seguretat.
- Transparència.
- Eficiència.
- Portabilitat.

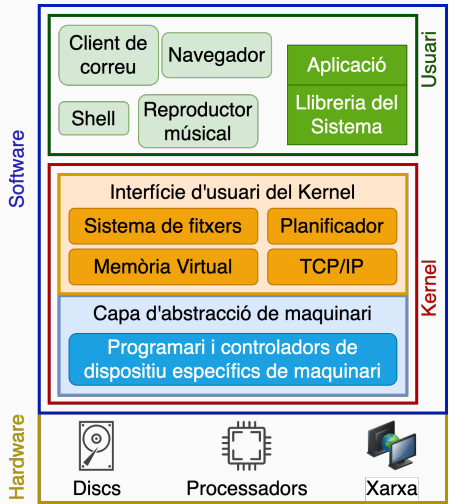


Figura 13: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què ofereix el sistema operatiu?

Què ofereix el sistema operatiu?

Serveis

- Controlar usuaris i aplicacions.
- Gestionar usuaris i aplicacions.
- Gestionar la Memòria.
- Sistema de fitxers.
- Planificadors.
- Eines de xarxa.

Garanties

- Seguretat.
- Transparència.
- Eficiència.
- Portabilitat.

Figura 13: Esquema d'elements i funcions d'un SO.

Què ofereix el sistema operatiu?

Serveis

- Controlar usuaris i aplicacions.
- Gestionar usuaris i aplicacions.
- Gestionar la Memòria.
- Sistema de fitxers.
- Planificadors.
- Eines de xarxa.

Garanties

- Seguretat.
- Transparència.
- Eficiència.
- Portabilitat.
- Estabilitat al llarg del temps.

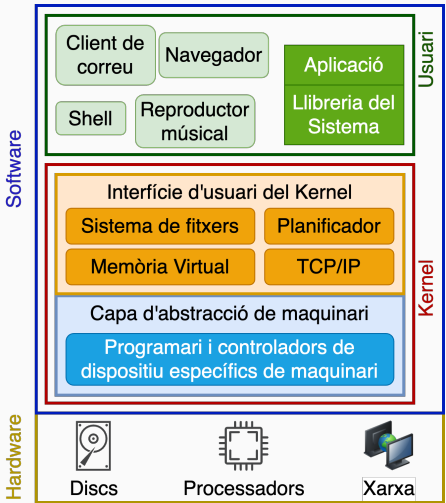


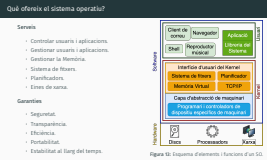
Figura 13: Esquema d'elements i funcions d'un SO.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

Introducció a la temàtica

Què ofereix el sistema operatiu?



Com s'organitza el sistema operatiu?

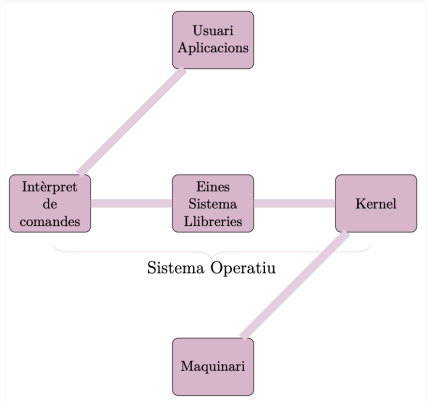


Figura 14: Organització del Sistema Operatiu

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└ Introducció a la temàtica

└ Com s'organitza el sistema operatiu?



Com s'organitza el sistema operatiu?

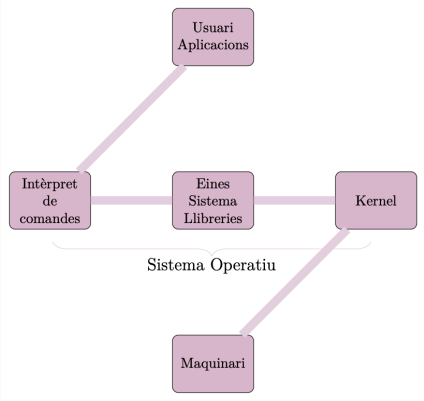


Figura 14: Organització del Sistema Operatiu

Maquina virtual

És la *visió* que té l'usuari del sistema operatiu durant una *sessió de treball*.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
└─ Introducció a la temàtica

└─ Com s'organitza el sistema operatiu?

Com s'organitza el sistema operatiu?

Diagrama de l'organització del sistema operatiu. El diagrama mostra la interacció entre els components d'un sistema operatiu:

- Usuari Aplicacions** (usuari) connectat amb **Intèrpret de comandes**.
- Intèrpret de comandes** connectat amb **Eines Sistema Llibreries**.
- Eines Sistema Llibreries** connectat amb **Kernel**.
- Kernel** connectat amb **Maquinari**.

Una línia horitzontal amb corbada connecta **Intèrpret de comandes**, **Eines Sistema Llibreries** i **Kernel**, etiquetada com a **Sistema Operatiu**.

Maquina virtual

És la visió que té l'usuari del sistema operatiu durant una sessió de treball.

Figura 14: Organització del Sistema Operatiu

Com s'organitza el sistema operatiu?

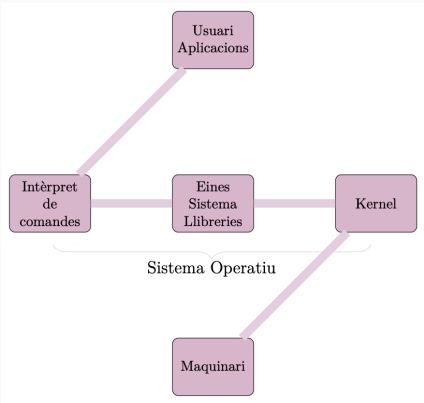


Figura 14: Organització del Sistema Operatiu

Maquina virtual

És la *visió* que té l'usuari del sistema operatiu durant una *sessió de treball*.

Dualitat

El sistema operatiu divideix el programari que té **tots els privilegis** (kernel) del programari que **no pot accedir a tots els recursos** (programes, llibreries, intèrpret de comandes,...).

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└ Introducció a la temàtica

└ Com s'organitza el sistema operatiu?

Com s'organitza el sistema operatiu?



Maquina virtual
És la visió que té l'usuari del sistema operatiu durant una sessió de treball.

Dualitat
El sistema operatiu divideix el programari que té **tots els privilegis** (kernel) del programari que **no pot accedir a tots els recursos** (programes, llibreries, intèrpret de comandes,...).

Figura 14: Organització del Sistema Operatiu

Què és una Màquina Virtual?

La **virtualització** presentar una visió abstracta dels recursos del sistema. Diversos processos **creuen** (tenen l'il·lusió) de disposar sempre d'un conjunt de recursos (màquinaria).

- **Simplicitat** ⇒ Il·lusió de propietat de recursos

Recordeu

Imagineu-vos que organitzem una LAN-party amb assistents, però només hi ha 1 pizza per alimentar-los. Com podem mostrar a cada un el seu propi tros de pizza i mantenir-los compromesos amb el nostre esdeveniment. Fàcil, creem 1 pizza virtual basada en la pizza real i donem a cada encarregat un tros de pizza virtual. Però, aquí ve un problema si tots els assistents mengen la pizza alhora, no funcionarà, alguns participants notarien el truc. Tanmateix, si programem els esdeveniments perquè molts participants s'ocupin de les activitats i no mengin la pizza, els organitzadors poden intercanviar en temps real la pizza real i la virtual a les taules on els participants descansen, perquè són els únics candidats a menjar i descobrir el truc, així que Si mengen, la pizza ha de ser real.

Java té un sandbox que permet utilitzar el Java independentment d'on executem.

VirtualBox i VMWare ens permeten executar sistemes operatius dins de sistemes operatius. Tenim un sistema operatiu amfitrió (màquina física) i un o més sistemes operatius virtuals (hostes). També és possible virtualitzar sobre el hardware sense la necessitat d'un sistema operatiu (amfitrió).

També tenim virtualització per contenidors (Docker).

Què és una Màquina Virtual?

La **virtualització** presentar una visió abstracta dels recursos del sistema. Diversos processos **creuen** (tenen l'il·lusió) de disposar sempre d'un conjunt de recursos (màquinaria).

- **Simplicitat** ➡ Il·lusió de propietat de recursos
- **Aïllament** ➡ Els bugs es donen en un entorn virtual i no físic.

Recordeu

Imagineu-vos que organitzem una LAN-party amb assistents, però només hi ha 1 pizza per alimentar-los. Com podem mostrar a cada un el seu propi tros de pizza i mantenir-los compromesos amb el nostre esdeveniment. Fàcil, creem 1 pizza virtual basada en la pizza real i donem a cada encarregat un tros de pizza virtual. Però, aquí ve un problema si tots els assistents mengen la pizza alhora, no funcionarà, alguns participants notarien el truc. Tanmateix, si programem els esdeveniments perquè molts participants s'ocupin de les activitats i no mengin la pizza, els organitzadors poden intercanviar en temps real la pizza real i la virtual a les taules on els participants descansen, perquè són els únics candidats a menjar i descobrir el truc, així que Si mengen, la pizza ha de ser real.

Java té un sandbox que permet utilitzar el Java independentment d'on executem.

VirtualBox i VMWare ens permeten executar sistemes operatius dins de sistemes operatius. Tenim un sistema operatiu amfitrió (màquina física) i un o més sistemes operatius virtuals (hostes). També és possible virtualitzar sobre el hardware sense la necessitat d'un sistema operatiu (amfitrió).

També tenim virtualització per contenidors (Docker).

- **Simplicitat** ➡ Il·lusió de propietat de recursos
- **Aïllament** ➡ Els bugs es donen en un entorn virtual i no físic.

Què és una Màquina Virtual?

La **virtualització** presentar una visió abstracta dels recursos del sistema. Diversos processos **creuen** (tenen l'il·lusió) de disposar sempre d'un conjunt de recursos (màquinaria).

- **Simplicitat** ➡ Il·lusió de propietat de recursos
- **Aïllament** ➡ Els bugs es donen en un entorn virtual i no físic.
- **Protecció** ➡ Els processos no es poden fer mal entre ells.

Recordeu

Imagineu-vos que organitzem una LAN-party amb assistents, però només hi ha 1 pizza per alimentar-los. Com podem mostrar a cada un el seu propi tros de pizza i mantenir-los compromesos amb el nostre esdeveniment. Fàcil, creem 1 pizza virtual basada en la pizza real i donem a cada encarregat un tros de pizza virtual. Però, aquí ve un problema si tots els assistents mengen la pizza alhora, no funcionarà, alguns participants notarien el truc. Tanmateix, si programem els esdeveniments perquè molts participants s'ocupin de les activitats i no mengin la pizza, els organitzadors poden intercanviar en temps real la pizza real i la virtual a les taules on els participants descansen, perquè són els únics candidats a menjar i descobrir el truc, així que Si mengen, la pizza ha de ser real.

Java té un sandbox que permet utilitzar el Java independentment d'on executem.

VirtualBox i VMWare ens permeten executar sistemes operatius dins de sistemes operatius. Tenim un sistema operatiu amfitrió (màquina física) i un o més sistemes operatius virtuals (hostes). També és possible virtualitzar sobre el hardware sense la necessitat d'un sistema operatiu (amfitrió).

També tenim virtualització per contenidors (Docker).

- **Simplicitat** ➡ Il·lusió de propietat de recursos
- **Aïllament** ➡ Els bugs es donen en un entorn virtual i no físic.
- **Protecció** ➡ Els processos no es poden fer mal entre ells.

Què és una Màquina Virtual?

La **virtualització** presentar una visió abstracta dels recursos del sistema. Diversos processos **creuen** (tenen l'il·lusió) de disposar sempre d'un conjunt de recursos (màquinaria).

- **Simplicitat** ⇒ Il·lusió de propietat de recursos
- **Aïllament** ⇒ Els bugs es donen en un entorn virtual i no físic.
- **Protecció** ⇒ Els processos no es poden fer mal entre ells.
- **Portabilitat** ⇒ Podem executar a totes les plataformes.

Recordeu

Imagineu-vos que organitzem una LAN-party amb assistents, però només hi ha 1 pizza per alimentar-los. Com podem mostrar a cada un el seu propi tros de pizza i mantenir-los compromesos amb el nostre esdeveniment. Fàcil, creem 1 pizza virtual basada en la pizza real i donem a cada encarregat un tros de pizza virtual. Però, aquí ve un problema si tots els assistents mengen la pizza alhora, no funcionarà, alguns participants notarien el truc. Tanmateix, si programem els esdeveniments perquè molts participants s'ocupin de les activitats i no mengin la pizza, els organitzadors poden intercanviar en temps real la pizza real i la virtual a les taules on els participants descansen, perquè són els únics candidats a menjar i descobrir el truc, així que Si mengen, la pizza ha de ser real.

Java té un sandbox que permet utilitzar el Java independentment d'on executem.

VirtualBox i VMWare ens permeten executar sistemes operatius dins de sistemes operatius. Tenim un sistema operatiu amfitrió (màquina física) i un o més sistemes operatius virtuals (hostes). També és possible virtualitzar sobre el hardware sense la necessitat d'un sistema operatiu (amfitrió).

També tenim virtualització per contenidors (Docker).

- **Simplicitat** ⇒ Il·lusió de propietat de recursos.
- **Aïllament** ⇒ Els bugs es donen en un entorn virtual i no físic.
- **Protecció** ⇒ Els processos no es poden fer mal entre ells.
- **Portabilitat** ⇒ Podem executar a totes les plataformes.

Què és la Memòria Virtual?

La *memòria virtual* permet que cada procés tingui la **il·lusió** que té accés exclusiu a **l'espai complet d'adreces de memòria del processador**. En realitat els processos utilitzen **diferents regions de la memòria** de l'ordinador, amb algunes regions traslladades al *disc* **si no hi ha prou memòria** per a tothom.

Aquesta **il·lusió** d'accés a tota la memòria la proporciona la unitat de **gestió de memòria (MMU)** d'un processador, que **tradueix** les *adreces virtuals* utilitzades pel programa en *adreces físiques* que representen ubicacions de memòria reals.

Què és la Memòria Virtual?

La memòria virtual permet que cada procés tingui la il·lusió que té accés exclusiu a l'espai complet d'adreces de memòria del processador. En realitat els processos utilitzen diferents regions de la memòria de l'ordinador, amb algunes regions traslladades al disc si no hi ha prou memòria per a tothom.

Aquesta il·lusió d'accés a tota la memòria la proporciona la unitat de gestió de memòria (MMU) d'un processador, que tradueix les adreces virtuals utilitzades pel programa en adreces físiques que representen ubicacions de memòria reals.

Il·lusionista

- **Interfície Simple per als Recursos Físics:** Ofereix una interfície simple i fàcil d'utilitzar per als recursos físics d'una màquina o sistema, ocultant la complexitat tècnica.

2024-09-30

Il·lusionista

- **Interfície Simple per als Recursos Físics:** Ofereix una interfície simple i fàcil d'utilitzar per als recursos físics d'una màquina o sistema, ocultant la complexitat tècnica.

Permet als usuaris utilitzar una impressora (*hardware*) sense conèixer els detalls tècnics d'aquesta impresora, com la interfície de comunicació, els controladors o els protocols de comunicació, disposem d'una funció (**imprimir**) que ens permet enviar un document a la impressora.

Il·lusionista

- **Interficie Simple per als Recursos Físics:** Ofereix una interfície simple i fàcil d'utilitzar per als recursos físics d'una màquina o sistema, ocultant la complexitat tècnica.

Permet als usuaris utilitzar una impressora (*hardware*) sense conèixer els detalls tècnics d'aquesta impressora, com la interfície de comunicació, els controladors o els protocols de comunicació, disposem d'una funció (**imprimir**) que ens permet enviar un document a la impressora.

Il·lusionista

- **Ús Exclusiu de Recursos:** Proporciona una abstracció que permet a una aplicació tenir ús exclusiu dels recursos quan sigui necessari, sense interferències d’altres aplicacions.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Quins són els rols del sistema operatiu? (II)

Il·lusionista

- **Ús Exclusiu de Recursos:** Proporciona una abstracció que permet a una aplicació tenir ús exclusiu dels recursos quan sigui necessari, sense interferències d’altres aplicacions.

Permet utilitzar un programa de videoconferència que utilitza la càmera i el micròfon i ens garantitza que cap altre programa pugui utilitzar-los al mateix temps.

2024-09-30

Il·lusionista

• **Ús Exclusiu de Recursos:** Proporciona una abstracció que permet a una aplicació tenir ús exclusiu dels recursos quan sigui necessari, sense interferències d’altres aplicacions.

Permet utilitzar un programa de videoconferència que utilitza la càmera i el micròfon i ens garantitza que cap altre programa pugui utilitzar-los al mateix temps.

Il·lusionista

- **Il·lusió d’Infinitat de Recursos de Maquinari:** Oferir una il·lusió d’infinitat per als recursos de maquinari, assegurant que les aplicacions puguin continuar funcionant sense problemes.

Il·lusionista

- **Il·lusió d’Infinitat de Recursos de Maquinari:** Oferir una il·lusió d’infinitat per als recursos de maquinari, assegurant que les aplicacions puguin continuar funcionant sense problemes.

Il·lusionista

- **Il·lusió d’Infinitat de Recursos de Maquinari:** Oferir una il·lusió d’infinitat per als recursos de maquinari, assegurant que les aplicacions puguin continuar funcionant sense problemes.
- Permet a un usuari tenir múltiples aplicacions obertes alhora, tot i que només una estigui en primer pla.

Il·lusionista

- **Il·lusió d’Infinitat de Recursos de Maquinari:** Oferir una il·lusió d’infinitat per als recursos de maquinari, assegurant que les aplicacions puguin continuar funcionant sense problemes.
- Permet a un usuari tenir múltiples aplicacions obertes alhora, tot i que només una estigui en primer pla.

Il·lusionista

- **Il·lusió d’Infinitat de Recursos de Maquinari:** Oferir una il·lusió d’infinitat per als recursos de maquinari, assegurant que les aplicacions puguin continuar funcionant sense problemes.
- Permet a un usuari tenir múltiples aplicacions obertes alhora, tot i que només una estigui en primer pla.
- Crea una il·lusió on cada procés creu que és propietari dels recursos hardware.

Il·lusionista

- **Il·lusió d’Infinitat de Recursos de Maquinari:** Oferir una il·lusió d’infinitat per als recursos de maquinari, assegurant que les aplicacions puguin continuar funcionant sense problemes.
- Permet a un usuari tenir múltiples aplicacions obertes alhora, tot i que només una estigui en primer pla.
- Crea una il·lusió on cada procés creu que és propietari dels recursos hardware.

Il·lusionista

- **Abstracció de Capacitats que no Estan Presentes Físicament:** Proporcionar abstraccions de capacitats que no estan físicament presents en el maquinari, com emuladors o màquines virtuals que executen sistemes operatius diferents.

Il·lusionista

• **Abstracció de Capacitats que no Estan Presentes Físicament:** Proporcionar abstraccions de capacitats que no estan físicament presents en el maquinari, com emuladors o màquines virtuals que executen sistemes operatius diferents.

Il·lusionista

- **Abstracció de Capacitats que no Estan Presentes Físicament:** Proporcionar abstraccions de capacitats que no estan físicament presents en el maquinari, com emuladors o màquines virtuals que executen sistemes operatius diferents.

Execució de múltiples sistemes operatius en una única màquina física. Per exemple, permet l'execució de Windows i Linux en una única màquina utilitzant diferents discs o particions.

Il·lusionista

- **Abstracció de Capacitats que no Estan Presentes Físicament:** Proporcionar abstraccions de capacitats que no estan físicament presents en el maquinari, com emuladors o màquines virtuals que executen sistemes operatius diferents.

Execució de múltiples sistemes operatius en una única màquina física. Per exemple, permet l'execució de Windows i Linux en una única màquina utilitzant diferents discs o particions.

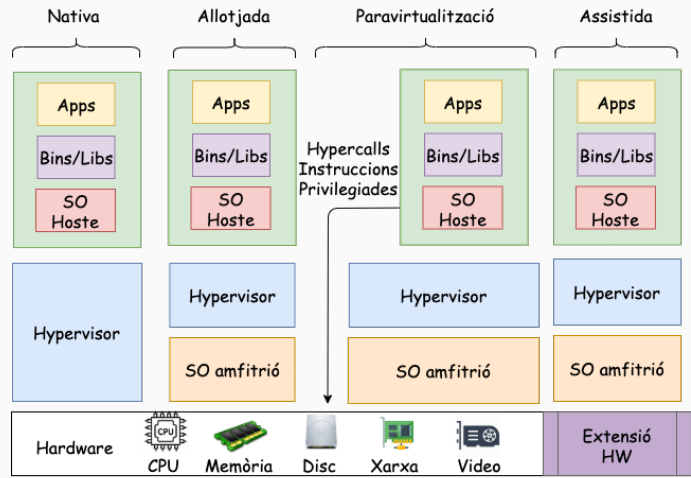


Figura 15: Esquema de tipus de virtualització de màquines virtuals

2024-09-30



- **Virtualització Nativa:** Hipervisor s'executa directament sobre el maquinari físic sense cap sistema operatiu intermedi. Exemples d'aquesta tecnologia inclouen VMware ESXi o Microsoft Hyper-V. Això ofereix un rendiment molt elevat, ja que no hi ha cap capa addicional entre el maquinari i l'hipervisor, ideal per a entorns de producció.
- **Virtualització Allotjada:** Hipervisor s'executa sobre un sistema operatiu ja existent. Per exemple, en KVM, s'utilitza Linux com a sistema operatiu base, i sobre aquest es creen les màquines virtuals.
- **Paravirtualització:** El sistema operatiu convidat sap que està sent virtualitzat i coopera amb l'hipervisor per millorar el rendiment. Un exemple destacat és Xen, que és molt utilitzat en entorns cloud per la seva eficiència.
- **Virtualització Assistida per Hardware:** Aquest model aprofita les capacitats del processador per accelerar el procés de virtualització. Processadors moderns com els d'Intel (VT-x) o AMD (AMD-V) ofereixen suport per a aquest tipus de virtualització (VMware o Hyper-V).

Àrbitre

- **Assignació de recursos:** Responsable de distribuir els recursos disponibles entre usuaris i aplicacions de manera eficient i justa.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Quins són els rols del sistema operatiu? (V)

Àrbitre

- **Assignació de recursos:** Responsable de distribuir els recursos disponibles entre usuaris i aplicacions de manera eficient i justa.

Àrbitre

- **Assignació de recursos:** Responsable de distribuir els recursos disponibles entre usuaris i aplicacions de manera eficient i justa.

Un sistema amb múltiples usuaris, el temps de processador s’ha de repartir de manera equitativa entre tots els usuaris que executen aplicacions.

2024-09-30

Àrbitre

- **Protecció/Aïllament:** Garanteix la segregació i la protecció d’usuaris i aplicacions.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Quins són els rols del sistema operatiu? (VI)

Àrbitre

- **Protecció/Aïllament:** Garanteix la segregació i la protecció d’usuaris i aplicacions.

Impedint que una aplicació bloquegi o afecti el funcionament d’altres aplicacions.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Quins són els rols del sistema operatiu? (VI)

Pega

- **Conjunt de Serveis Comuns:** Proporciona un conjunt de serveis i funcionalitats comunes que poden ser compartits i reutilitzats per diverses parts d'un sistema. **Compartició:** Simplifica SI s'assumeixen sempre les mateixes primitives bàsiques. **Reutilització:** Evita torna a implementar funcionalitats comunes. Permet evolucionar de forma independent els components.

Pega

• **Conjunt de Serveis Comuns:** Proporciona un conjunt de serveis i funcionalitats comunes que poden ser compartits i reutilitzats per diverses parts d'un sistema. **Compartició:** Simplifica SI s'assumeixen sempre les mateixes primitives bàsiques. **Reutilització:** Evita torna a implementar funcionalitats comunes. Permet evolucionar de forma independent els components.

Pega

- **Conjunt de Serveis Comuns:** Proporciona un conjunt de serveis i funcionalitats comunes que poden ser compartits i reutilitzats per diverses parts d'un sistema. **Compartició:** Simplifica SI s'assumeixen sempre les mateixes primitives bàsiques. **Reutilització:** Evita torna a implementar funcionalitats comunes. Permet evolucionar de forma independent els components.

El **sistema operatiu** ens ofereix un **sistema de fitxers**. Aquest servei gestiona la disposició física dels fitxers a l'emmagatzematge, controla l'accés als fitxers i ofereix una interfície per interactuar amb ells. A més a més, ens **proporciona lliberies i components** que permet als **desenvolupadors** utilitzar aquest servei de forma **transparent**. *(read, write, open, close, ...)*

2024-09-30

Pega

• **Conjunt de Serveis Comuns:** Proporciona un conjunt de serveis i funcionalitats comunes que poden ser compartits i reutilitzats per diverses parts d'un sistema. **Compartició:** Simplifica SI s'assumeixen sempre les mateixes primitives bàsiques. **Reutilització:** Evita torna a implementar funcionalitats comunes. Permet evolucionar de forma independent els components.

El **sistema operatiu** ens ofereix un **sistema de fitxers**. Aquest servei gestiona la disposició física dels fitxers a l'emmagatzematge, controla l'accés als fitxers i ofereix una interfície per interactuar amb ells. A més a més, ens **proporciona lliberies i components** que permet als **desenvolupadors** utilitzar aquest servei de forma **transparent**. *(read, write, open, close, ...)*

Anàlisi: Què fa aquest programa? (I)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Anàlisi: Què fa aquest programa? (I)

2024-09-30

Anàlisi: Què fa aquest programa? (I)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

Anàlisi: Què fa aquest programa? (I)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

./prog H

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Anàlisi: Què fa aquest programa? (I)

Anàlisi: Què fa aquest programa? (I)

./prog H

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

Anàlisi: Què fa aquest programa? (I)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

./prog H

H H ...

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
└─ Introducció a la temàtica

└─ Anàlisi: Què fa aquest programa? (I)



Anàlisi: Què fa aquest programa? (I)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

./prog H

H H ...

./prog H & ./prog 0

???

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
└─ Introducció a la temàtica

└─ Anàlisi: Què fa aquest programa? (I)



Anàlisi: Què fa aquest programa? (II)

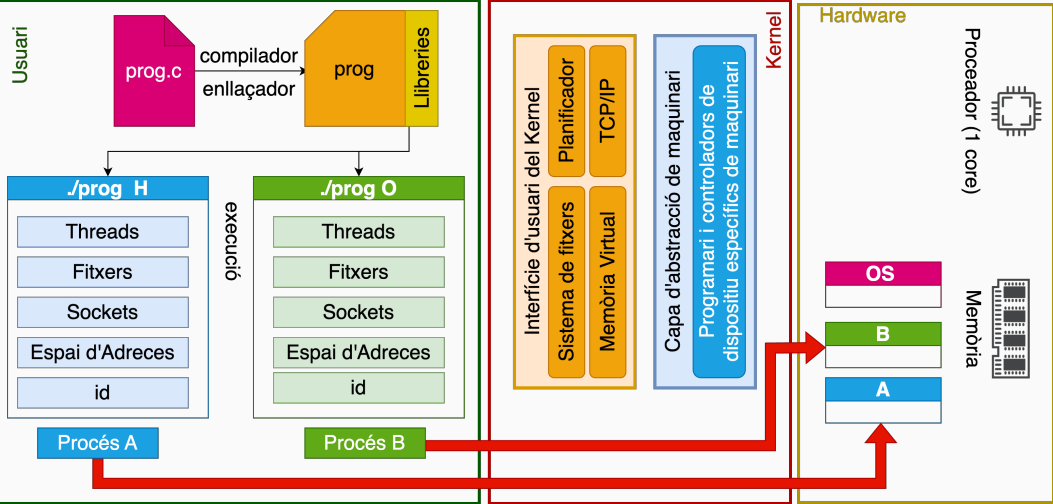
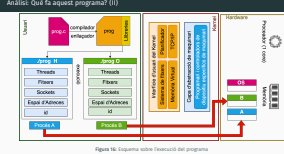


Figura 16: Esquema sobre l'execució del programa

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
└ Introducció a la temàtica

└ Anàlisi: Què fa aquest programa? (II)



En primer lloc, el programa c s'ha de compilar amb un compilador i enllaçar amb les llibreries del sistema que ens proporciona el sistema operatiu.

Un cop tenim un executable, si l'executem el sistema operatiu crearà una estructura anomenada procés on es guardarà un identificador, un espai de memòria,... aquesta estructura és una de les abstraccions essencials per la funció d'il·lusionisme.

Fixeu-vos que el mateix programa pot ser executat dos cops i el sistema operatiu generarà 2 processos diferents. Amb identificadors diferents i espai de memòria diferent.

Anàlisi: Què fa aquest programa? (III)

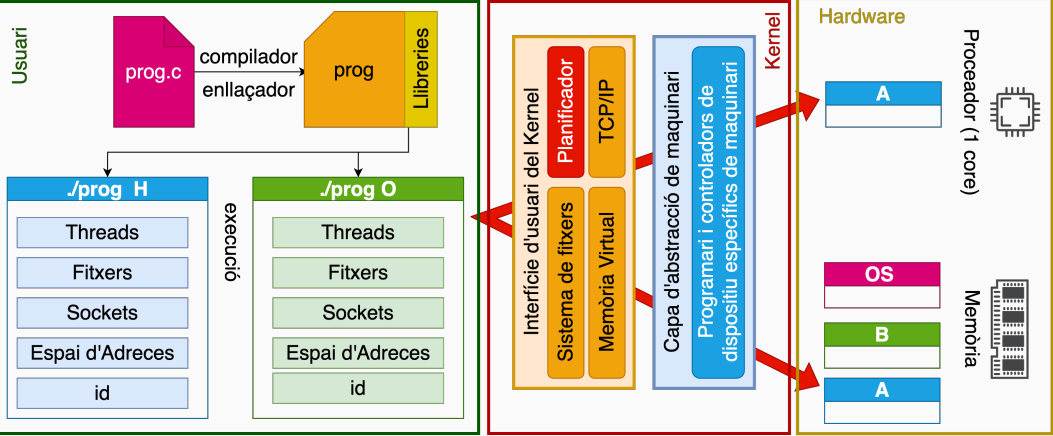


Figura 17: Esquema sobre l'execució del programa

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
Introducció a la temàtica

Anàlisi: Què fa aquest programa? (III)

Per tant, el sistema operatiu utilitzant el planificador assignarà el processador (assumim que només tenim 1 core) a 1 procés durant un temps determinat i anirà intercanviant (canvi de context) aquest procés cada X temps, d'aquesta manera cada procés creu tenir tots els recursos per ells (il·lusió).

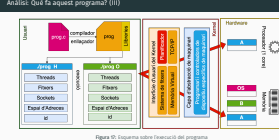


Figura 10: Esquema sobre l'execució del programa

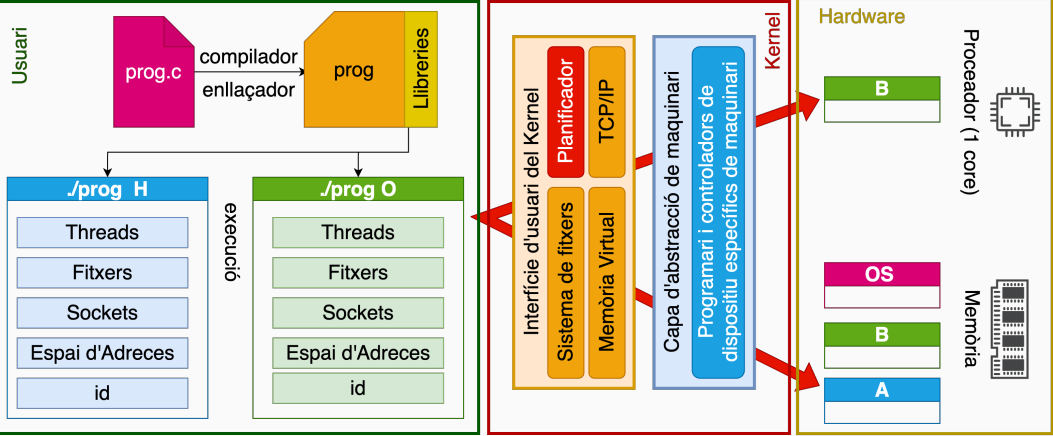


Figura 18: Esquema sobre l'execució del programa

2024-09-30

Per fer-ho, el sistema operatiu guardà una foto de l'estat actual del procés a memòria i la recuperarà més endavant quan li torni a donar dret d'execució.

Noteu també que s'ha creat en memòria una subregió independent per cada procés. Quan un procés intenta accedir una zona de memòria forà de la seva subregió o inexistent es dona el famós SEGMENTATION FAULT.

Com en l'últim exemple on el procés intenta accedir a una zona de memòria que no li pertany. Això és una de les funcions de protecció que ens ofereix el sistema operatiu. Al llarg del curs veurem amb més detall com funcionen tots aquests mecanismes.



Figura 18: Esquema sobre l'execució del programa

Anàlisi: Què fa aquest programa? (V)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Anàlisi: Què fa aquest programa? (V)

2024-09-30

Anàlisi: Què fa aquest programa? (V)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

Anàlisi: Què fa aquest programa? (V)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

Depenen de la prioritat dels processos **A** o **B** poden tenir més temps de *CPU* que l'altre i sortir els missatges de forma desordenada.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└ Introducció a la temàtica

└ Anàlisi: Què fa aquest programa? (V)

Anàlisi: Què fa aquest programa? (V)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

Depenen de la prioritat dels processos **A** o **B** poden tenir més temps de *CPU* que l'altre i sortir els missatges de forma desordenada.

Anàlisi: Què fa aquest programa? (V)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

Depenen de la prioritat dels processos **A** o **B** poden tenir més temps de *CPU* que l'altre i sortir els missatges de forma desordenada.

./prog H & ./prog O
H H H ... H O H ... H O O ...

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
└─ Introducció a la temàtica

└─ Anàlisi: Què fa aquest programa? (V)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

Depenen de la prioritat dels processos **A** o **B** poden tenir més temps de *CPU* que l'altre i sortir els missatges de forma desordenada.
./prog H & ./prog O
H H H ... H O H ... H O O ...

Anàlisi: Què fa aquest programa? (VI)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

```
./prog & ; ./prog 0
```

?????

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
└─ Introducció a la temàtica

└─ Anàlisi: Què fa aquest programa? (VI)

```
./prog & ; ./prog 0
mmmm

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

Anàlisi: Què fa aquest programa? (VII)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    while(1)
    {
        printf("%s\n", argv[1]);
    }
    return 0
}
```

./prog & ; ./prog 0

Segmentation Fault 0 0 ...

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└ Introducció a la temàtica

└ Anàlisi: Què fa aquest programa? (VII)



Anàlisi: Què fa aquest programa? (VIII)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int *p = malloc(sizeof(int));
    printf("(%d) %p\n", getpid(), p);
    return 0
}
```

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└ Introducció a la temàtica

└ Anàlisi: Què fa aquest programa? (VIII)

Anàlisi: Què fa aquest programa? (VIII)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int *p = malloc(sizeof(int));
    printf("(%d) %p\n", getpid(), p);
    return 0
}
```

Anàlisi: Què fa aquest programa? (VIII)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int *p = malloc(sizeof(int));
    printf("(%d) %p\n", getpid(), p);
    return 0
}
```

./prog1

```
(611) p: 0x5570014a02a0
(611) p: 1
(611) p: 2
(611) p: 3
(611) p: 4
```

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└ Introducció a la temàtica

└ Anàlisi: Què fa aquest programa? (VIII)

Anàlisi: Què fa aquest programa? (VIII)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int *p = malloc(sizeof(int));
    printf("(%d) %p\n", getpid(), p);
    return 0
}
```

./prog1

```
(611) p: 0x5570014a02a0
(611) p: 1
(611) p: 2
(611) p: 3
(611) p: 4
```


Anàlisi: Què fa aquest programa? (VIII)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int *p = malloc(sizeof(int));
    printf("(d) %p\n",getpid(),p);
    return 0
}
```

./prog1

```
(611) p: 0x5570014a02a0
(611) p: 1
(611) p: 2
(611) p: 3
(611) p: 4
```

./prog1 & ./prog1

????

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
└─ Introducció a la temàtica

└─ Anàlisi: Què fa aquest programa? (VIII)



Anàlisi: Què fa aquest programa? (IX)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int *p = malloc(sizeof(int));
    printf("(%d) %p\n", getpid(), p);
    return 0
}
```

./prog1 & ./prog1

```
(611) p: 0x5570014a02a0
(612) p: 0x5570014a02a0
(611) p: 1
(612) p: 1
(611) p: 2
(612) p: 2
```

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└ Introducció a la temàtica

└ Anàlisi: Què fa aquest programa? (IX)

Anàlisi: Què fa aquest programa? (IX)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int *p = malloc(sizeof(int));
    printf("(%d) %p\n", getpid(), p);
    return 0
}
```

./prog1 & ./prog1

```
(611) p: 0x5570014a02a0
(612) p: 0x5570014a02a0
(611) p: 1
(612) p: 1
(611) p: 2
(612) p: 2
```

Anàlisi: Què fa aquest programa? (X)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int *p = malloc(sizeof(int));
    printf("(%d) %p\n", getpid(), p);
    return 0
}
```

Les **adreces virtuals** protegeixen els processos entre ells i permeten que tots dos processos puguin fer servir la mateixa adreça sense afectar-se.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└ Introducció a la temàtica

└ Anàlisi: Què fa aquest programa? (X)

Anàlisi: Què fa aquest programa? (X)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int *p = malloc(sizeof(int));
    printf("(%d) %p\n", getpid(), p);
    return 0
}
```

Les **adreces virtuals** protegeixen els processos entre ells i permeten que tots dos processos puguin fer servir la mateixa adreça sense afectar-se.

- Complexitat de la **programació distribuïda** (*concurrent* i *paral·lela*).

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Reptes en el disseny dels sistemes operatius (I)

- Complexitat de la **programació distribuïda** (*concurrent i paral·lela*).
- Complexitat pel **context** (*mòbil, IoT, servidors, centres de dades, ...*).

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Reptes en el disseny dels sistemes operatius (I)

- Complexitat de la **programació distribuïda** (*concurrent i paral·lela*).
- Complexitat pel **context** (*mòbil, IoT, servidors, centres de dades, ...*).

- Complexitat de la **programació distribuïda** (*concurrent i paral·lela*).
- Complexitat pel **context** (*mòbil, IoT, servidors, centres de dades, ...*).
- Complexitat per la gran varietat d'elements de maquinari (**heterogeneïtat**).

- Complexitat de la **programació distribuïda** (*concurrent i paral·lela*).
- Complexitat pel **context** (*mòbil, IoT, servidors, centres de dades, ...*).
- Complexitat per la gran varietat d'elements de maquinari (**heterogeneïtat**).
- Complexitat en la **portabilitat** i la **compatibilitat**.

Reptes en el disseny dels sistemes operatius (I)

- Complexitat de la **programació distribuïda** (*concurrent i paral·lela*).
- Complexitat pel **context** (*mòbil, IoT, servidors, centres de dades, ...*).
- Complexitat per la gran varietat d'elements de maquinari (**heterogeneïtat**).
- Complexitat en la **portabilitat** i la **compatibilitat**.
- Equilibri entre **funcionalitat** i **rendiment**.

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Reptes en el disseny dels sistemes operatius (I)

2024-09-30

Reptes en el disseny dels sistemes operatius (I)

- Complexitat de la **programació distribuïda** (concurrent i paral·lela).
- Complexitat pel **context** (mòbil, IoT, servidors, centres de dades, ...).
- Complexitat per la gran varietat d'elements de maquinari (**heterogeneïtat**).
- Complexitat en la **portabilitat** i la **compatibilitat**.
- Equilibri entre **funcionalitat** i **rendiment**.

- Complexitat de la **programació distribuïda** (*concurrent i paral·lela*).
- Complexitat pel **context** (*mòbil, IoT, servidors, centres de dades, ...*).
- Complexitat per la gran varietat d'elements de maquinari (**heterogeneïtat**).
- Complexitat en la **portabilitat** i la **compatibilitat**.
- Equilibri entre **funcionalitat** i **rendiment**.
- Equilibri entre **rendiment** i ús d'energia.

- Complexitat de la **programació distribuïda** (concurrent i paral·lela).
- Complexitat pel **context** (mòbil, IoT, servidors, centres de dades, ...).
- Complexitat per la gran varietat d'elements de maquinari (**heterogeneïtat**).
- Complexitat en la **portabilitat** i la **compatibilitat**.
- Equilibri entre **funcionalitat** i **rendiment**.
- Equilibri entre **rendiment** i ús d'energia.

- **Maximitzar la fiabilitat:** Els sistemes han de fer el que estan dissenyats per fer en tots els casos, fins i tot en cas d'errors inesperats.

- **Maximitzar la fiabilitat:** Els sistemes han de fer el que estan dissenyats per fer en tots els casos, fins i tot en cas d'errors inesperats.
- **Maximitzar la disponibilitat:** Els sistemes han d'estar disponibles per a l'ús quan els usuaris ho necessiten, minimitzant el temps d'aturada i reparació causat per fallades.

- **Maximitzar la fiabilitat:** Els sistemes han de fer el que estan dissenyats per fer en tots els casos, fins i tot en cas d'errors inesperats.
- **Maximitzar la disponibilitat:** Els sistemes han d'estar disponibles per a l'ús quan els usuaris ho necessiten, minimitzant el temps d'aturada i reparació causat per fallades.
- **Seguretat:** Els sistemes han de protegir-se contra accions malicioses i accidents involuntaris.

- **Maximitzar la fiabilitat:** Els sistemes han de fer el que estan dissenyats per fer en tots els casos, fins i tot en cas d'errors inesperats.
- **Maximitzar la disponibilitat:** Els sistemes han d'estar disponibles per a l'ús quan els usuaris ho necessiten, minimitzant el temps d'aturada i reparació causat per fallades.
- **Seguretat:** Els sistemes han de protegir-se contra accions malicioses i accidents involuntaris.

Imagineu el **sistema operatiu** d'un *vehicle*. Aquest sistema ha de garantir que el *vehicle* no es pugui **controlar de forma remota** per un atacant. També ha de ser **tolerant a fallades** i **recuperar-se d'errors** sense posar en perill la seguretat dels *passatgers*.

- **Maximitzar la fiabilitat:** Els sistemes han de fer el que estan dissenyats per fer en tots els casos, fins i tot en cas d'errors inesperats.
- **Maximitzar la disponibilitat:** Els sistemes han d'estar disponibles per a l'ús quan els usuaris ho necessiten, minimitzant el temps d'aturada i reparació causat per fallades.
- **Seguretat:** Els sistemes han de protegir-se contra accions malicioses i accidents involuntaris.

Imagineu el **sistema operatiu** d'un *vehicle*. Aquest sistema ha de garantir que el *vehicle* no es pugui **controlar de forma remota** per un atacant. També ha de ser **tolerant a fallades** i **recuperar-se d'errors** sense posar en perill la seguretat dels *passatgers*.

- **Escalables:** Els sistemes han de funcionar bé quan s’afegeixen recursos (usuaris, processos, ...) o quan es redueixen.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Reptes en el disseny dels sistemes operatius (III)

- **Escalables:** Els sistemes han de funcionar bé quan s’afegeixen recursos (usuaris, processos, ...) o quan es redueixen.
- **Mantenibles:** Els sistemes han de ser fàcils de mantenir i evolucionar al llarg del temps.

- **Escalables:** Els sistemes han de funcionar bé quan s’afegeixen recursos (usuaris, processos, ...) o quan es redueixen.
- **Mantenibles:** Els sistemes han de ser fàcils de mantenir i evolucionar al llarg del temps.

El **sistema operatiu** d’un telèfon mòbil ha de ser **escalable** perquè el nombre d’usuaris pot augmentar molt ràpidament. També ha de ser **mantenible** perquè els usuaris esperen actualitzacions periòdiques del **sistema operatiu**. A més a més, aquestes *actualitzacions* s’han de poder *instal·lar* de forma *transparent i sense afectar el funcionament del telèfon*.

- **Escalables:** Els sistemes han de funcionar bé quan s’afegeixen recursos (usuaris, processos, ...) o quan es redueixen.
- **Mantenibles:** Els sistemes han de ser fàcils de mantenir i evolucionar al llarg del temps.

El **sistema operatiu** d’un telèfon mòbil ha de ser **escalable** perquè el nombre d’usuaris pot augmentar molt ràpidament. També ha de ser **mantenible** perquè els usuaris esperen actualitzacions periòdiques del **sistema operatiu**. A més a més, aquestes actualitzacions s’han de poder *instal·lar* de forma *transparent i sense afectar el funcionament del telèfon*.

- Els sistemes operatius són la base dels sistemes informàtics. (IoT, Servidors, Mòbils, PC, ...).

En particular, sovint passa que un canvi en la tecnologia fa que alguns idea obsoleta i ràpidament desapareix. No obstant això, un altre canvi tecnològic podria tornar-lo a reviure. Això és especialment cert quan el canvi té a veure amb el rendiment relatiu de les diferents parts del sistema. Per exemple, quan les CPUs es van fer molt més ràpides que les Memòries les caches van gaunyar molta importància, però que passa si en un futur les memòries són molt més ràpides que les CPUs? En aquest cas, les caches ja no serien necessàries. En biologia, l'extinció és per sempre, però en informàtica, de vegades només és per uns quants anys.

- Els sistemes operatius són la base dels sistemes informàtics. (IoT, Servidors, Mòbils, PC, ...).
- Els sistemes operatius ens permeten entendre com funciona *realment* un sistema informàtic.

En particular, sovint passa que un canvi en la tecnologia fa que alguns idea obsoleta i ràpidament desapareix. No obstant això, un altre canvi tecnològic podria tornar-lo a reviure. Això és especialment cert quan el canvi té a veure amb el rendiment relatiu de les diferents parts del sistema. Per exemple, quan les CPUs es van fer molt més ràpides que les Memòries les caches van gaunyar molta importància, però que passa si en un futur les memòries són molt més ràpides que les CPUs? En aquest cas, les caches ja no serien necessàries. En biologia, l'extinció és per sempre, però en informàtica, de vegades només és per uns quants anys.

- Els sistemes operatius són la base dels sistemes informàtics. (IoT, Servidors, Mòbils, PC, ...).
- Els sistemes operatius ens permeten entendre com funciona *realment* un sistema informàtic.
- Els conceptes bàsics dels sistemes operatius són aplicables a altres sistemes i problemes quotidians (**planificació de tasques**).

En particular, sovint passa que un canvi en la tecnologia fa que alguns idea obsoleta i ràpidament desapareix. No obstant això, un altre canvi tecnològic podria tornar-lo a reviure. Això és especialment cert quan el canvi té a veure amb el rendiment relatiu de les diferents parts del sistema. Per exemple, quan les CPUs es van fer molt més ràpides que les Memòries les caches van gaunyar molta importància, però que passa si en un futur les memòries són molt més ràpides que les CPUs? En aquest cas, les caches ja no serien necessàries. En biologia, l'extinció és per sempre, però en informàtica, de vegades només és per uns quants anys.

- Els sistemes operatius són la base dels sistemes informàtics. (IoT, Servidors, Mòbils, PC, ...).
- Els sistemes operatius ens permeten entendre com funciona *realment* un sistema informàtic.
- Els conceptes bàsics dels sistemes operatius són aplicables a altres sistemes i problemes quotidians (**planificació de tasques**).
- Optimitzar el rendiment de les aplicacions.

En particular, sovint passa que un canvi en la tecnologia fa que alguns idea obsoleta i ràpidament desapareix. No obstant això, un altre canvi tecnològic podria tornar-lo a reviure. Això és especialment cert quan el canvi té a veure amb el rendiment relatiu de les diferents parts del sistema. Per exemple, quan les CPUs es van fer molt més ràpides que les Memòries les caches van gaunyar molta importància, però que passa si en un futur les memòries són molt més ràpides que les CPUs? En aquest cas, les caches ja no serien necessàries. En biologia, l'extinció és per sempre, però en informàtica, de vegades només és per uns quants anys.

- Els sistemes operatius són la base dels sistemes informàtics. (IoT, Servidors, Mòbils, PC, ...).
- Els sistemes operatius ens permeten entendre com funciona *realment* un sistema informàtic.
- Els conceptes bàsics dels sistemes operatius són aplicables a altres sistemes i problemes quotidians (**planificació de tasques**).
- Optimitzar el rendiment de les aplicacions.

Per què estudiar Sistemes Operatius?

- Els sistemes operatius són la base dels sistemes informàtics. (IoT, Servidors, Mòbils, PC, ...).
- Els sistemes operatius ens permeten entendre com funciona *realment* un sistema informàtic.
- Els conceptes bàsics dels sistemes operatius són aplicables a altres sistemes i problemes quotidians (**planificació de tasques**).
- Optimitzar el rendiment de les aplicacions.
- Compendre les vulnerabilitats, proteccions i mitigacions de riscos de seguretat.

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Per què estudiar Sistemes Operatius?

En particular, sovint passa que un canvi en la tecnologia fa que alguns idea obsoleta i ràpidament desapareix. No obstant això, un altre canvi tecnològic podria tornar-lo a reviure. Això és especialment cert quan el canvi té a veure amb el rendiment relatiu de les diferents parts del sistema. Per exemple, quan les CPUs es van fer molt més ràpides que les Memòries les caches van gaunyar molta importància, però que passa si en un futur les memòries són molt més ràpides que les CPUs? En aquest cas, les caches ja no serien necessàries. En biologia, l'extinció és per sempre, però en informàtica, de vegades només és per uns quants anys.

2024-09-30

Per què estudiar Sistemes Operatius?

- Els sistemes operatius són la base dels sistemes informàtics. (IoT, Servidors, Mòbils, PC, ...).
- Els sistemes operatius ens permeten entendre com funciona *realment* un sistema informàtic.
- Els conceptes bàsics dels sistemes operatius són aplicables a altres sistemes i problemes quotidians (**planificació de tasques**).
- Optimitzar el rendiment de les aplicacions.
- Compendre les vulnerabilitats, proteccions i mitigacions de riscos de seguretat.

Per què estudiar Sistemes Operatius?

- Els sistemes operatius són la base dels sistemes informàtics. (IoT, Servidors, Mòbils, PC, ...).
- Els sistemes operatius ens permeten entendre com funciona *realment* un sistema informàtic.
- Els conceptes bàsics dels sistemes operatius són aplicables a altres sistemes i problemes quotidians (**planificació de tasques**).
- Optimitzar el rendiment de les aplicacions.
- Compendre les vulnerabilitats, proteccions i mitigacions de riscos de seguretat.

La comprensió en les decisions de disseny dels sistemes operatius i el raonament sobre els pros/contra us permetrà rescatar idees obsoletes que poden ser útils en el futur dels nous sistemes informàtics.

En particular, sovint passa que un canvi en la tecnologia fa que alguns idea obsoleta i ràpidament desapareix. No obstant això, un altre canvi tecnològic podria tornar-lo a reviure. Això és especialment cert quan el canvi té a veure amb el rendiment relatiu de les diferents parts del sistema. Per exemple, quan les CPUs es van fer molt més ràpides que les Memòries les caches van gaunyar molta importància, però que passa si en un futur les memòries són molt més ràpides que les CPUs? En aquest cas, les caches ja no serien necessàries. En biologia, l'extinció és per sempre, però en informàtica, de vegades només és per uns quants anys.

- Els **sistemes operatius** estan presents en tots els dispositius que fem servir.

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Conclusions

- Els **sistemes operatius** estan presents en tots els dispositius que fem servir.
- El *disseny* dels **sistemes operatius** es **complex** i requereix integrar dispositius diferents (*forma,espai,temps*).

2024-09-30

Unitat 0 · Sistemes Operatius (SO)

└─ Introducció a la temàtica

└─ Conclusions

Conclusions

- Els **sistemes operatius** estan presents en tots els dispositius que fem servir.
- El *disseny* dels **sistemes operatius** es **complex** i requereix integrar dispositius diferents (*forma,espai,temps*).

- Els **sistemes operatius** estan presents en tots els dispositius que fem servir.
- El *disseny* dels **sistemes operatius** es **complex** i requereix integrar dispositius diferents (*forma,espai,temps*).
- Els sistema operatius ens proporciona una il·lusió d’una màquina virtual infinita.

- Els **sistemes operatius** estan presents en tots els dispositius que fem servir.
- El *disseny* dels **sistemes operatius** es **complex** i requereix integrar dispositius diferents (*forma,espai,temps*).
- Els sistema operatius ens proporciona una il·lusió d’una màquina virtual infinita.

- Els **sistemes operatius** estan presents en tots els dispositius que fem servir.
- El *disseny* dels **sistemes operatius** es **complex** i requereix integrar dispositius diferents (*forma,espai,temps*).
- Els sistema operatius ens proporciona una il·lusió d’una màquina virtual infinita.
- El **sistema operatiu** és *il·lusionista, árbrit* i proporciona un conjunt de serveis comuns per permetre la interacció entre programari i maquinari.

- Els **sistemes operatius** estan presents en tots els dispositius que fem servir.
- El *disseny* dels **sistemes operatius** es **complex** i requereix integrar dispositius diferents (*forma,espai,temps*).
- Els sistema operatius ens proporciona una il·lusió d’una màquina virtual infinita.
- El **sistema operatiu** és *il·lusionista, árbrit* i proporciona un conjunt de serveis comuns per permetre la interacció entre programari i maquinari.

- Els **sistemes operatius** estan presents en tots els dispositius que fem servir.
- El *disseny* dels **sistemes operatius** es **complex** i requereix integrar dispositius diferents (*forma,espai,temps*).
- Els sistema operatius ens proporciona una il·lusió d’una màquina virtual infinita.
- El **sistema operatiu** és *il·lusionista, árbrit* i proporciona un conjunt de serveis comuns per permetre la interacció entre programari i maquinari.
- Un **sistema operatiu** té cura del **rendiment, seguretat, portabilitat i fiabilitat**.

- Els **sistemes operatius** estan presents en tots els dispositius que fem servir.
- El *disseny* dels **sistemes operatius** es **complex** i requereix integrar dispositius diferents (*forma,espai,temps*).
- Els **sistema operatius** ens proporciona una il·lusió d’una màquina virtual infinita.
- El **sistema operatiu** és *il·lusionista, árbrit* i proporciona un conjunt de serveis comuns per permetre la interacció entre programari i maquinari.
- Un **sistema operatiu** té cura del **rendiment, seguretat, portabilitat i fiabilitat**.

PREGUNTES?

Materials del curs

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

TAKE HOME MESSAGE: Els **sistemes operatius** ajuden als programadors a desenvolupar programari robust de forma independent del maquinari.



Figura 19: Això és tot per avui

2024-09-30

Unitat 0 · Sistemes Operatius (SO)
└ Introducció a la temàtica

└ Això és tot per avui

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: Els sistemes operatius ajuden als programadors a desenvolupar programari robust de forma independent del maquinari.




Figura 19: Això és tot per avui