

• ANTES DEL VIDEO:

Software: Parte lógica del computador

Sistema Operativo

↓
Loaders

Bios → Programas utilitarios → Drivers

• DESPUÉS DE VER EL VIDEO

- 1) ¿Tiene sentido enseñar SO?
- 2) ¿Qué resulta de utilidad?
- 3) ¿Peso mucho que otra?

Notas importantes:

Parte del software comienza sobre la máquina

multiplexación: Combina señales y transmitir

- M. recursos del hardware

- parte en software de otros

basic structure

[A] [A] [Daemon]

Virtual Memory, Scheduling, File System,
Device Drivers, IPC, Network Stack, etc

[Core] [Core] [Core] [Core] [Core]

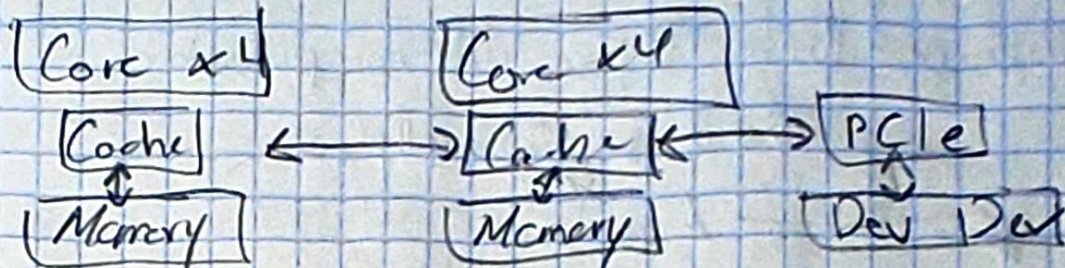
[Memory]

↓
Memory compartida y ya

Peligrosa idea?

Linux works:

(Peligroso porque no es así)



Sistema no es en sí el sistema, es un componente

- Arquitectura ('no magnament')
- Hardware design

Problema

Limitaciones del S.O., sellando funcionalidad por dentro, pero lo que mejor algunos que son necesarios.

Encontrar sistemas operativos no tiene mucho sentido, ya que en sí, es más importante la caja que el sistema que responde al entorno.

Real problema: S.O. por todo lo que lleve
S.O. un S.O.

Novelty

El espacio y desarrollo permite una nueva vista con Linux, específicamente, guiar en como este sistema, no busca en sí el sistema sino todo que trae para usarlo.

Pensamiento

2) Utilidad

/ Genial que sea complex! Eso es lo que hacemos

OS People \ / Architecture People

Difícil comunicación → Describen situaciones que ambos no ven

¿Que sería de utilidad?

Parte de entender los cambios del diseño, involucra una trayectoria registrada. Es por ello que derivamos fuerzas, pero al hacer esto perdemos la utilidad de la situación, en el caso sistema operativo le quitamos volúmenes a la pregunta 1, queriendo cambiar la forma en la que vemos el SO como solo uno de todos, ignorando!

Machine learning, Hardware, DB, Graph Processing, Security, Privacy, etc, reestructurar el SO!

Como vemos, la utilidad radica quiz, de igual manera, como mencionaba, Linux se enfoca en buscar esa dirección de trabajo.

3) ¿Pero muerto?

Creemos que el pero muerto proviene de un modelo de hacer SO antiguo... es por eso que la prioridad radica en un modelo que adapte a lo que hoy somos capaces de realizar