

# Sistemas Operativos

Adín Ramírez

`adin.ramirez@mail.udp.cl`

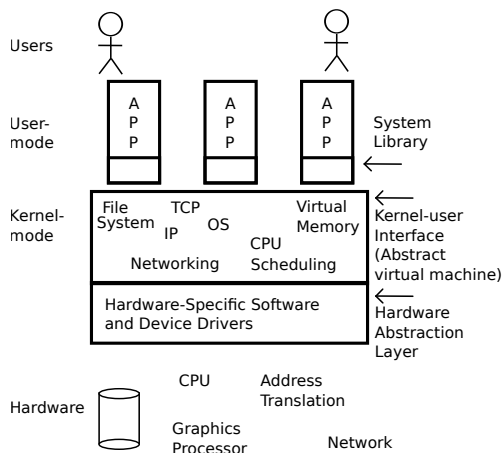
Sistemas Operativos (CIT2003-1)  
1er. Semestre 2015

# Objetivos de la clase

- Definición de un sistema operativo
  - ▶ Software que administra los recursos de una computadora para sus usuarios y aplicaciones
- Retos SO
  - ▶ Confiabilidad, seguridad, responsabilidad (tiempos de respuesta), portabilidad, etc.
- Breve historia de SO
  - ▶ ¿Cómo se relaciona OS X, Windows, y Linux?

# ¿Qué es un sistema operativo?

- Es el software que administra los recursos de una computadora para sus usuarios y aplicaciones



# Roles del sistema operativo

## ■ Árbitro

- ▶ Reserva de recursos entre usuarios, y aplicaciones
- ▶ Aislamiento de diferentes usuarios y aplicaciones (unos de otros)
- ▶ Comunicación entre usuarios y aplicaciones

## ■ Ilusionista

- ▶ Cada aplicación cree que tiene la máquina completa para ella
- ▶ Número de procesadores infinitos, (casi) memoria infinita, almacenamiento confiable, transporte de red confiable

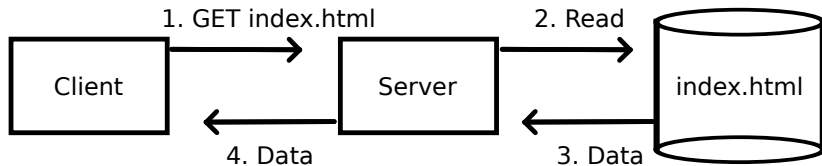
## ■ Pegamento

- ▶ Librerías, interfaz de usuarios

# Pregunta interesante

- ¿Qué necesitamos del hardware para poder
  - ▶ aislar diferentes aplicaciones unas de otras?
  - ▶ aislar el acceso de diferentes usuarios a las aplicaciones de los otros?

## Ejemplo: web service



- ¿Cómo administra el servidor múltiples peticiones simultaneas?
- ¿Cómo mantenemos al cliente seguro de spyware embebido en los scripts del sitio web?
- ¿Cómo mantenemos las actualizaciones del sitio web consistentes?

# Retos de SO

## ■ Confiabilidad

- ▶ ¿El sistema hace para lo que fue diseñado?
- ▶ Disponibilidad
  - ¿Qué porción del tiempo está el sistema trabajando?
  - Mean time to failure (MTTF)
  - Mean time to repair (MTTR)

## ■ Seguridad

- ▶ ¿Puede el sistema ser comprometido por un atacante?
- ▶ Privacidad
  - Los datos son accesibles solo por usuarios autorizados

## ■ Ambos necesitan un cuidadoso **diseño** y **codificación**

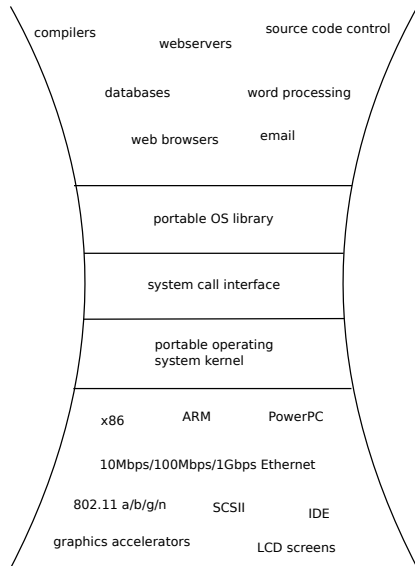
# Más retos

## ■ Portabilidad

- ▶ Para los programas
  - Application programming interface (API)
  - Interfaz abstracta de máquina

## ■ Para el sistema operativo

- ▶ Capa de abstracción de hardware
- ▶ E.g., Pintos provee rutinas del kernel específicas para el hardware



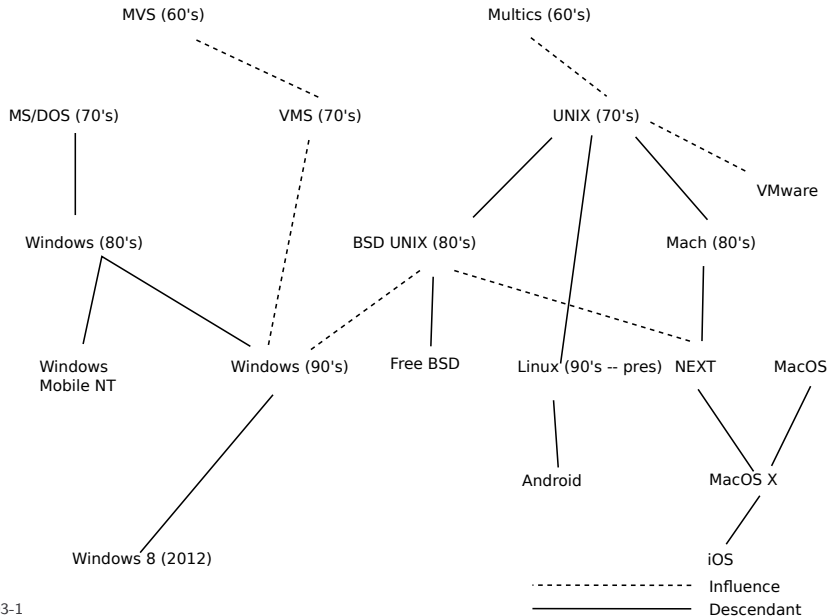


# Aún más retos

## ■ Rendimiento

- ▶ Latencia o tiempo de respuesta
  - ¿Cuánto tiempo tarda una operación en completarse?
- ▶ Throughput (caudal)
  - ¿Cuántas operaciones pueden hacerse por unidad de tiempo?
- ▶ Overhead (sobre carga)
  - ¿Cuánto trabajo extra es realizado por el SO?
- ▶ Justicia
  - ¿Cuán equitativo es el rendimiento recibido por distintos usuarios?
- ▶ Predecibilidad
  - ¿Qué tan consistente es el rendimiento en el tiempo?

# Historia de los SO



# Rendimiento de las computadoras en el tiempo

	1981	1996	2011	factor
MIPS <sup>1</sup>	1	300	10000	10K
MIPS/\$	\$100K	\$ 30	\$ 0.5	200K
DRAM	128 KB	128 MB	10 GB	100K
Disk	10 MB	4 GB	1 TB	100K
Home Internet	9.6 Kbps	256 Kbps	5 Mbps	500
LAN network	3 Mbps (shared)	10 Mbps	1 Gbps	300
Users per machine	100	1	≪ 1	100+

<sup>1</sup>Microprocessor without Interlocked Pipelines Stages

# Primeros sistemas operativos

Los computadores eran caros

- Una aplicación a la vez
  - ▶ Tenía control total sobre el hardware
  - ▶ SO era una librería en tiempo de ejecución
  - ▶ Los usuarios hacían fila para usar el computador
- Sistemas por batch
  - ▶ Mantenían el CPU ocupado utilizando una cola de trabajos
  - ▶ SO cargaría el siguiente trabajo mientras ejecutaba uno
  - ▶ Usuarios enviaban tareas, y esperaban, y esperaban, y ...

# Sistemas Operativos de tiempo compartido

Computadoras y personas son caras

- Múltiples usuarios en una computadora al mismo tiempo
  - ▶ Multi programación: ejecuta varios programas al mismo tiempo
  - ▶ Rendimiento interactivo: tratar de completar el trabajo de todos rápidamente
  - ▶ Conforme las computadoras se hicieron más baratas, se trata de optimizar para el tiempo del usuario y no de la computadora

# Sistemas Operativos de hoy

Computadoras son baratas

- Smartphones
- Sistemas embebidos
- Web servers
- Laptops
- Tablets
- Máquinas virtuales
- ...

# Sistemas Operativos del futuro

- Data centers gigantes
- Aumento del número de procesadores por computadora
- Aumento de número de computadoras por usuario
- Almacenamiento de *gran* escala

# Pregunta interesante

- ¿Cómo debería un sistema operativo asignar tiempo de procesamiento entre dos usuarios que compiten?
  - ▶ Dar CPU al primero que llegó
  - ▶ Al que necesite la menor cantidad de recursos para terminar
  - ▶ Al que necesite más recursos para terminar
  - ▶ ¿Qué pasa si necesitamos asignar memoria?
  - ▶ ¿O espacio de disco?



# Puntos a recordar

- Sistema operativo: administra recursos para usuarios y aplicaciones
- Roles: árbitro, ilusionista, pegamento
- Retos del SO
  - ▶ Confiabilidad
  - ▶ Seguridad
  - ▶ Portabilidad
  - ▶ Rendimiento

# Libro de texto

## Consejos

- El texto es sofisticado
- No lo van a absorber todo en la primera pasada
- Lean antes de cada clase
- Lean después de cada clase
- No dejen las lecturas para después, ni mucho menos para días antes de la solemne