



Sistemas Operativos y Distribuidos



Práctico 1 Introducción

- 1.1.** Definir el concepto de Sistema Operativo (SO). ¿Cuáles son sus principales objetivos?
- 1.2.** Clasificar los sistemas operativos de acuerdo a la administración de tareas, usuarios y recursos. Dar ejemplos de sistemas operativos que hayas usado y que estén incluidos en alguna de estas clasificaciones.
- 1.3.** Definir el concepto de Internet de las cosas (abreviado en inglés IoT). ¿Considerás que es necesario que electrodomésticos como el televisor, el lavarropas, el microondas, la cámara de fotos o un cepillo de dientes tengan incorporado un sistema operativo? ¿O podrían utilizarse sin un sistema operativo?
- 1.4.** Describir algunos de los desafíos que surgen de diseñar sistemas operativos para dispositivos móviles en comparación con el diseño de sistemas operativos para PCs tradicionales.
- 1.5.** Definir brevemente las propiedades esenciales de los siguientes tipos de sistemas operativos:
 - a) Lote (*Batch*)
 - b) Interactivo (*Interactive*)
 - c) Tiempo compartido (*Time sharing*)
 - d) Tiempo real (*Real Time*)
 - e) Red (*Network*)
 - f) Distribuido (*Distributed*)
 - g) Clúster (*Cluster*)
 - h) Móvil (*Handheld*)
- 1.6.** Identificar las ventajas y desventajas de los sistemas operativos *open-source*. Incluir qué clases de personas, empresas u organizaciones puedan llegar a considerar cada aspecto como una ventaja o una desventaja.
- 1.7.** ¿Qué es el *kernel* (núcleo) de un sistema operativo? ¿En qué área de la memoria reside?
- 1.8.** Especificar cada una de las estructuras de datos utilizadas en el *kernel* y dar ejemplos de dónde podrían usarse cada una de ellas.

1.9. A principios del año 1992 hubo un intercambio de mails entre Andrew Tanenbaum y Linus Torvalds, el cual se conoció como: *Tanenbaum/Linus "Linux is obsolete" debates*. Dicho debate estuvo centrado en el concepto de *"microkernel"*. Puntualizar ventajas y desventajas del mismo así como también del kernel monolítico.

1.10. ¿Qué entendés por multiprogramación? ¿Cuál es la principal motivación para implementar esta técnica?

1.11. ¿Existe alguna dependencia entre el sistema operativo y la arquitectura subyacente? En caso de que la respuesta sea afirmativa, citar ejemplos en los que:

- a) El *hardware* haya influido en el diseño de un sistema operativo.
- b) El sistema operativo haya influido en el diseño del *hardware*.

1.12. ¿Cuál es el propósito de las interrupciones? ¿Cuál es la diferencia entre un *trap* y una interrupción? ¿Podría un programa de usuario generar un *trap*?

1.13. El acceso directo a memoria (DMA) se utiliza en dispositivos de I/O de alta velocidad con el fin de evitar el aumento en la carga de ejecución de la CPU.

- a) ¿Cómo se coordina la transferencia entre la CPU y el dispositivo?
- b) ¿Cómo sabe la CPU que las operaciones de memoria han sido completadas?
- c) Dado que se permite a la CPU ejecutar otros programas mientras que el controlador DMA está transfiriendo datos; ¿puede interferir con la ejecución de los programas de usuario? Si es así, describir qué tipos de interferencia causa.

1.14. Describir las diferencias entre multiprocesamiento simétrico y asimétrico. ¿Cuáles son las ventajas y desventajas de un sistema con múltiples procesadores?

1.15. ¿Por qué son necesarios múltiples modos de operación? ¿Cómo determina el sistema operativo en qué modo se encuentra?

1.16. ¿Por qué deben ser privilegiadas las siguientes instrucciones?

- a) Modificar el contenido del timer.
- b) Cambiar a modo kernel.
- c) Deshabilitar interrupciones.
- d) Acceso a dispositivo de E/S.

1.17. Investigar acerca de la diferencia entre los conceptos de protección y seguridad.

1.18. ¿Cuál es el propósito de una llamada a sistema (*system call*)?

1.19. Relacionar los conceptos de *system call* y API (*Application Programming Interface*) que forman parte de un sistema operativo.

1.20. ¿Cuál es el objetivo del intérprete de comandos (*Shell*)? ¿Por qué generalmente está separado del kernel?

1.21 ¿Cuál es la principal ventaja del diseño de un sistema operativo por capas? ¿Cuáles son sus desventajas?

1.22. ¿Cuáles son las ventajas de usar módulos cargables en el núcleo (*loadable kernel modules*)?

1.23. Distinguir entre los modelos cliente-servidor y los *peer-to-peer* para sistemas distribuidos.

1.24. ¿Qué entendés por *cloud computing*? Describir los siguientes términos:

- a) *Infrastructure as a service* (IaaS)
- b) *Platform as a service* (PaaS)
- c) *Software as a service* (SaaS)

1.25. Desde tu punto de vista, ¿el cómputo en la nube representa una ventaja o una desventaja con respecto al modelo de cómputo tradicional?