Sistemas Operativos Avanzados

Profesor:

Dr. J. Octavio Gutiérrez García

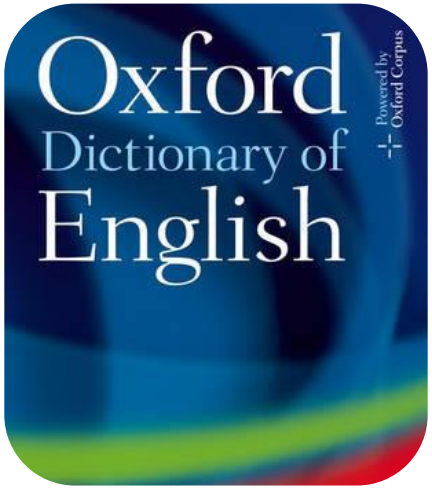
octavio.gutierrez@itam.mx

Objetivos

- Analizar temas avanzados de sistemas operativos, y

- Revisar críticamente resultados de investigación del área.

Definición de “avanzado”



- …having the most recently developed ideas, methods, etc.
- …at a high or difficult level

Temario

1) Introducción a los sistemas operativos
2) Sincronización y Concurrencia
3) Planificación
4) Administración de la Memoria Virtual
5) Sistemas de archivos
6) Virtualización
7) Diseño de sistemas operativos

Temario

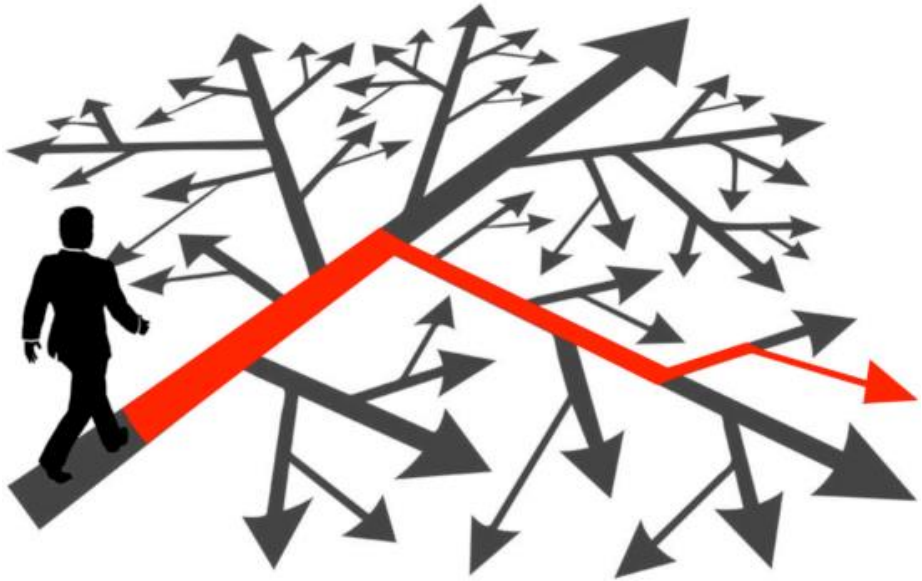
Sincronización y Concurrencia

- FlexSC: Flexible system call scheduling with exception-less system calls *
- Scheduler Activations: Effective Kernel Support for the User-Level Management of Parallelism *
- Eraser: A Dynamic Data Race Detector for Multithreaded Programs *
- Threads Cannot be Implemented as a Library
- Eliminating Receive Livelock in an Interrupt-Driven Kernel *
- Experience with processes and monitors in Mesa

Temario

Planificación

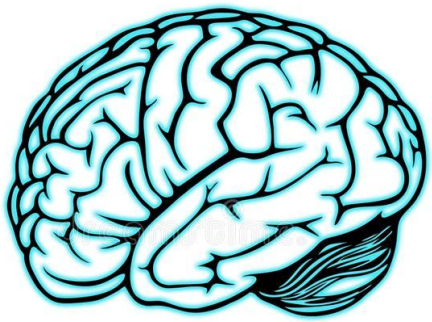
- Multi-core real-time scheduling for generalized parallel task models *
- Lottery scheduling: flexible proportional-share resource management
- The Linux Scheduler: a decade of wasted cores *
- The Spring Kernel: A new paradigm for real-time systems
- A fair share scheduler

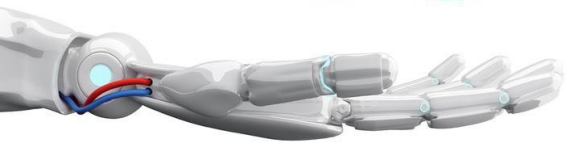


Temario

Administración de la Memoria Virtual

- Practical, transparent operating system support for superpages *
- Memory Resource Management in VMware ESX Server *
- Native Client: a sandbox for portable, untrusted x86 native code





Temario

Sistemas de archivos

- The Google File System *
- Rethink the Sync *
- Leases: An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency
- Venti: a new approach to archival storage
- ShieldFS: A Self-healing, Ransomware-aware Filesystem
- Design and Implementation of the Second Extended Filesystem
- Journaling the Linux ext2fs Filesystem
- Bigtable: A Distributed Storage System for Structured Data

Temario

Virtualización

- Xen and the Art of Virtualization *
- A comparison of software and hardware techniques for x86 virtualization
- Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors
- Containers and virtual machines at scale: A comparative study

Temario

Diseño de sistemas operativos

- LegoOS: A Disseminated, Distributed OS for Hardware Resource Disaggregation
- Hints for Computer System Design

Evaluación

- 10 críticas verbales en clase (a lo más una por artículo) **10%**
- Una exposición de un artículo de investigación frente a grupo de 60 minutos (+/- 5) **20%**
- Ensayo tipo “survey” de un tema avanzado de sistemas operativos **25%**
  Analizar **20 o más** artículos (sin tomar en cuenta referencias de introducción).
  Primera fecha de entrega opcional: 20 de Octubre de 2022 (sin calificación)
  Segunda fecha de entrega obligatoria: 27de Noviembre de 2022 (con calificación)
- Implementación (o simulación) visual interactiva de un mecanismo selecto * de sistemas operativos avanzados **20%**
  Fecha de entrega: 28 de Noviembre de 2022. Hora por Definir.
- Examen final teórico **25%**

Todo reporte deberá estar escrito con sus propias palabras

Exposiciones

	Integrantes	Artículo	Fecha aproximada de Exposición
Equipo 1	VILLEGAS JUAREZ DIEGO VILLANUEVA LOPEZ JOSE ANDRES	Threads Cannot be Implemented as a Library	1° de Septiembre
Equipo 2	CAMBA ALMONACI MITZI YAEL GOMEZ ESQUIVEL PABLO EDUARDO	Practical, transparent operating system support for superpages	6 de Octubre*
Uni-equipo 3	GALINDO AÑEL VIANEY JERUSALEN	Venti: a new approach to archival storage	27 de Octubre*
Equipo 4	MARTINEZ GAYTAN RENATA GARCIA GONZALEZ SALVADOR	Container-based operating system virtualization / Containers and virtual machines at scale	17 de Noviembre
Equipo 5	SANTOS DE LA CRUZ MARCOS TONATIUH ONTIVEROS TANUS HEID JORGE ALAN	LegoOS: A Disseminated, Distributed OS for Hardware Resource Disaggregation	24 de Noviembre

Artículos científicos

- Adams, K., and Agesen, O. "*A comparison of software and hardware techniques for x86 virtualization*" ACM Sigplan Notices, 41, 11, 2006, 2-13.

- Anderson, T. E., Bershad, B. N., Lazowska, E. D., Levy, H. M. "*Scheduler activations: effective kernel support for the user-level management of parallelism*", In Proceedings of the thirteenth ACM symposium on Operating systems principles. ACM, New York, NY, USA, 1991, 95-109.

Artículos científicos

- Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A., "*Xen and the art of virtualization*" In Proceedings of the nineteenth ACM symposium on Operating systems principles, ACM, New York, NY, USA, 2003, pp. 164-177.

- Boehm, H.J. "*Threads cannot be implemented as a library*". SIGPLAN Not. 40, 6, 2005, 261-268.

- Card, R., Ts'o T., and Tweedie, S. "*Design and Implementation of the Second Extended Filesystem*" Proc. of First Dutch International Symposium on Linux, ISBN 90-367-0385-9, 1994.

Artículos científicos

- Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., and Gruber, R. E., "*Bigtable: a distributed storage system for structured data*" In Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI '06), Vol. 7. USENIX Association, Berkeley, CA, USA, 2006, pp. 1-15.

- Continella, A., Guagnelli, A., Zingaro, G., De Pasquale, G., Barenghi, A., Zanero, S., & Maggi, F. (2016, December). *ShieldFS: a self-healing, ransomware-aware filesystem*. In Proceedings of the 32nd Annual Conference on Computer Security Applications (pp. 336-347). ACM

- Ghemawat, S., Gobioff, H., and Leung, S.-T, "*The google file system*". In proceedings of the nineteenth ACM symposium on Operating systems principles (New York, NY, USA, 2003), ACM, pp. 29–43.

Artículos científicos

- Gray C., and Cheriton, D. "*Leases: an efficient fault-tolerant mechanism for distributed file cache consistency*" SIGOPS Oper. Syst. Rev. 23, 5, 1989, pp. 202-210.

- Kay J., and Lauder, P., "*A fair share scheduler*" Commun. ACM 31 (1), 1988, 44-55.

- Lampson B. W., and Redell, D.D. "*Experience with processes and monitors in Mesa*" Commun. ACM 23, 2, 1980, 105-117.

Artículos científicos

- Lampson, B.W. "*Hints for Computer Systems Design*" In Proceedings of the Ninth ACM Symposium on Operating Systems Principles, USA, 1983, pp. 33-48.

- Lozi, J. P., Lepers, B., Funston, J., Gaud, F., Quéma, V., & Fedorova, A. (2016, April). *The Linux scheduler: a decade of wasted cores*. In Proceedings of the Eleventh European Conference on Computer Systems (p. 1). ACM

- Mogul, J. C., and Ramakrishnan, K. K. "*Eliminating receive livelock in an interrupt-driven kernel*". ACM Trans. Comput. Syst. 15, 3, 1997, 217-252.

Artículos científicos

- Navarro, J., Iyer, S., Druschel, P., and Cox, A. "*Practical, transparent operating system support for superpages*". In Proceedings of the 5th symposium on Operating systems design and implementation, ACM, New York, NY, USA, 2002, pp. 89-104.

- Nightingale, E. B., Veeraraghavan, K., Chen, P. M., and Flinn, J. "*Rethink the sync*" In Proceedings of the 7th symposium on Operating systems design and implementation (OSDI '06). USENIX Association, Berkeley, CA, USA, 2006, 1-14.

- Quinlan S., and Dorward S. "*Venti: A New Approach to Archival Storage*". In Proceedings of the Conference on File and Storage Technologies (FAST '02), Darrell D. E. Long (Ed.). USENIX Association, Berkeley, CA, USA, 2002, pp. 89-101.

Artículos científicos

- Shan, Y., Huang, Y., Chen, Y., & Zhang, Y. (2018). *LegoOS: A Disseminated, Distributed {OS} for Hardware Resource Disaggregation*. In 13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18) (pp. 69-87)

- Soares, L., & Stumm, M. (2010, October). "*FlexSC: Flexible system call scheduling with exception-less system calls*". In Proceedings of the 9th USENIX conference on Operating systems design and implementation (pp. 33-46). USENIX Association.

Artículos científicos

- Saifullah, A., Li, J., Agrawal, K., Lu, C., & Gill, C. (2013). "*Multi-core real-time scheduling for generalized parallel task models*". Real-Time Systems, 49(4), 404-435.

- Soltesz, S., Pötzl, H., Fiuczynski, M. E., Bavier, A., & Peterson, L. (2007, March). "*Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors*". In ACM SIGOPS Operating Systems Review (Vol. 41, No. 3, pp. 275-287). ACM.

Artículos científicos

- Savage, S., Burrows, M., Nelson, G., Sobalvarro, P., and Anderson, T, "*Eraser: a dynamic data race detector for multithreaded programs*" ACM Trans. Comput. Syst. 15(4), 1997, 391-411.

- Stankovic, J.A., and Krithi R. "*The spring kernel: A new paradigm for real-time systems*" IEEE Software, 8, 3, 1991, 62-72.

- Tweedie, S. C. "*Journaling the Linux ext2fs Filesystem*" In proceedings of the 4th Annual LinuxExpo, Durham, NC., 1998.

Artículos científicos

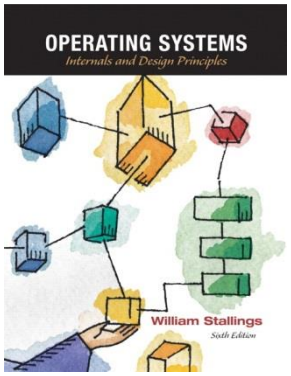
- Waldspurger, C.A., "*Memory resource management in VMware ESX server*". SIGOPS Oper. Syst. Rev. 36, SI, 2002, pp. 181-194.

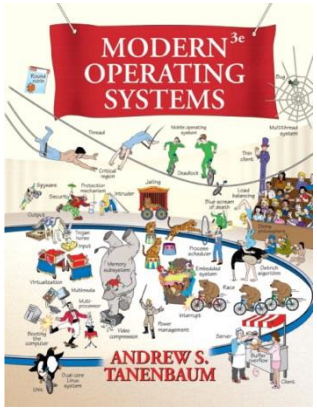
- Waldspurger C. A., and Weihl, W. E., "*Lottery scheduling: flexible proportional-share resource management*" In Proceedings of the 1st USENIX conference on Operating Systems Design and Implementation, USENIX Association, Berkeley, CA, USA, 1994, Article 1.

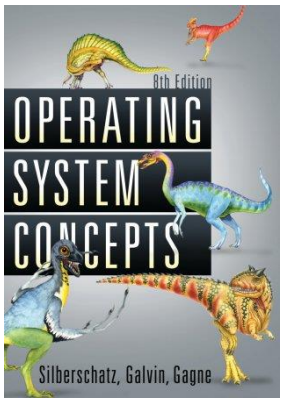
- Yee, B., Sehr, D., Dardyk, G., Chen J. B., Muth, R., Ormandy, T., Okasaka, S., Narula, N., and Fullagar, N., "*Native Client: a sandbox for portable, untrusted x86 native code*" Commun. ACM 53, 1, 91-99, 2010.

Bibliografía

- William Stallings, *Operating Systems: Internals and Design Principles*, Prentice Hall, 6th edition, 2009.
- Andrew S. Tanenbaum, *Modern Operating Systems*, Prentice Hall; 3rd. edition, 2007.
- Silberschatz, Galvin, and Gagne, *Operating Systems Concepts*, Wiley; 8th. edition (July 29, 2008)

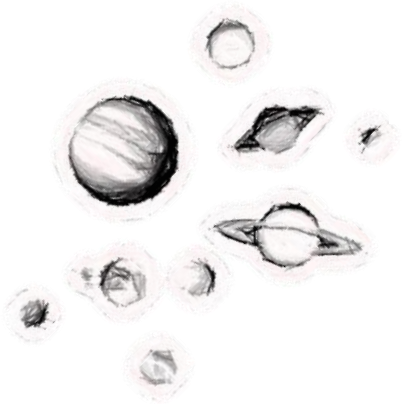






Vínculos web

- Stanford's Advanced Topics in Operating Systems Course, http://www.stanford.edu/class/cs240/
- CMU's Advanced and Distributed Operating Systems Course, http://www.cs.cmu.edu/~15712/
- Cornell's Advanced Course in Computer Systems, http://www.cs.cornell.edu/courses/cs614/2003sp/
- Xen project, http://www.xen.org
- KVM project, http://www.linux-kvm.org/page/Main_Page
- D.C. Marinescu, "Cloud Computing and Computer Clouds", Lecture Notes, Computer Science Division, Department of Electrical Engineering & Computer Science, University of Central Florida, 2012, https://www.cs.ucf.edu/~dcm/Teaching/CDA5532-CloudComputing/LectureNotes.pdf



Sistemas Operativos Avanzados

Tema 1. **Introducción a los sistemas operativos**

1

GO Mobile

2


Mac
Powered By
GNU/Linux
bada
clearOS
android
symbian
OS
palm
Java
CentOS

gc cloud

globus online

LEGION

Grid OS

openstack™

cloudstack

Windows Azure™

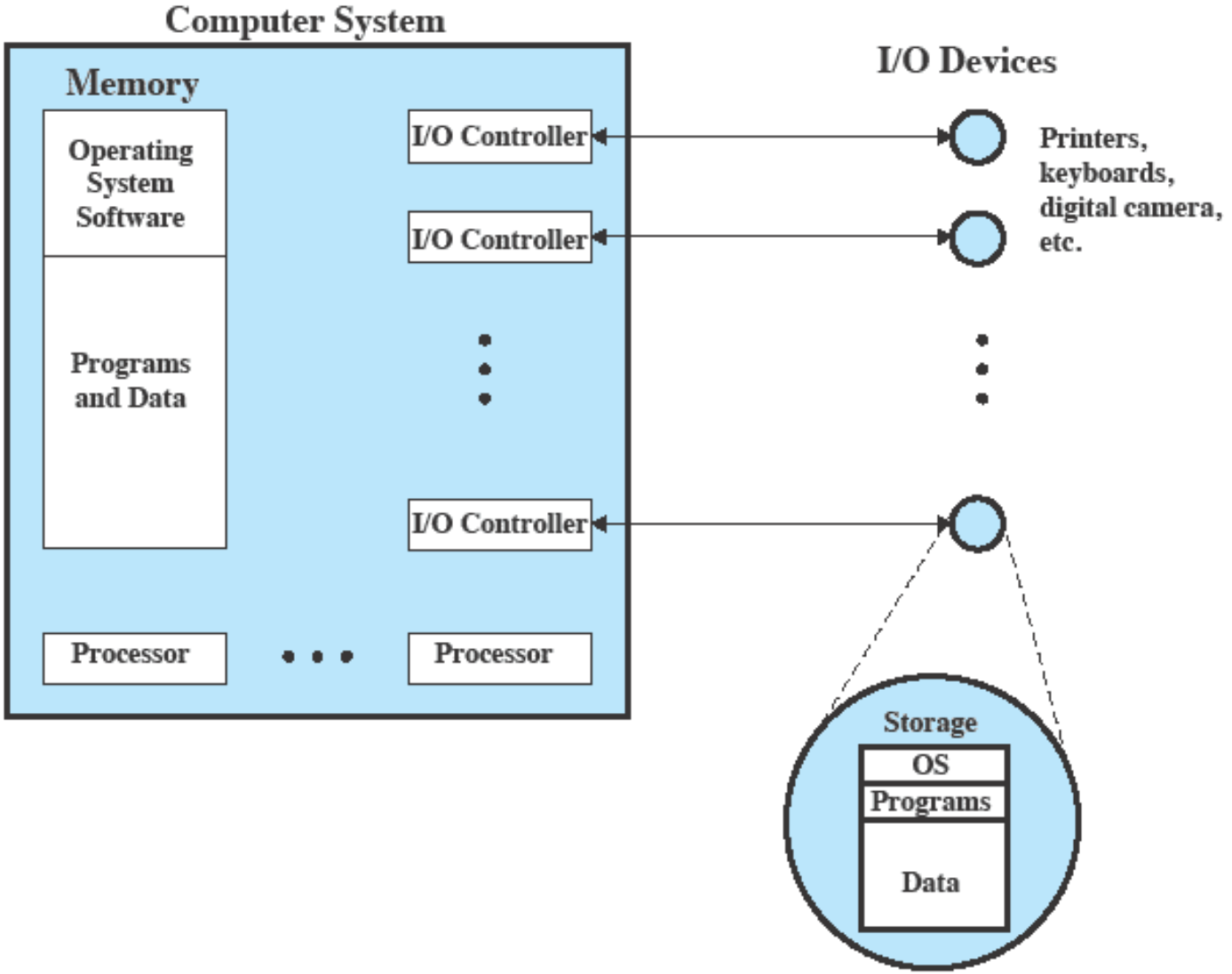
EUCALYPTUS

OpenNebula.org

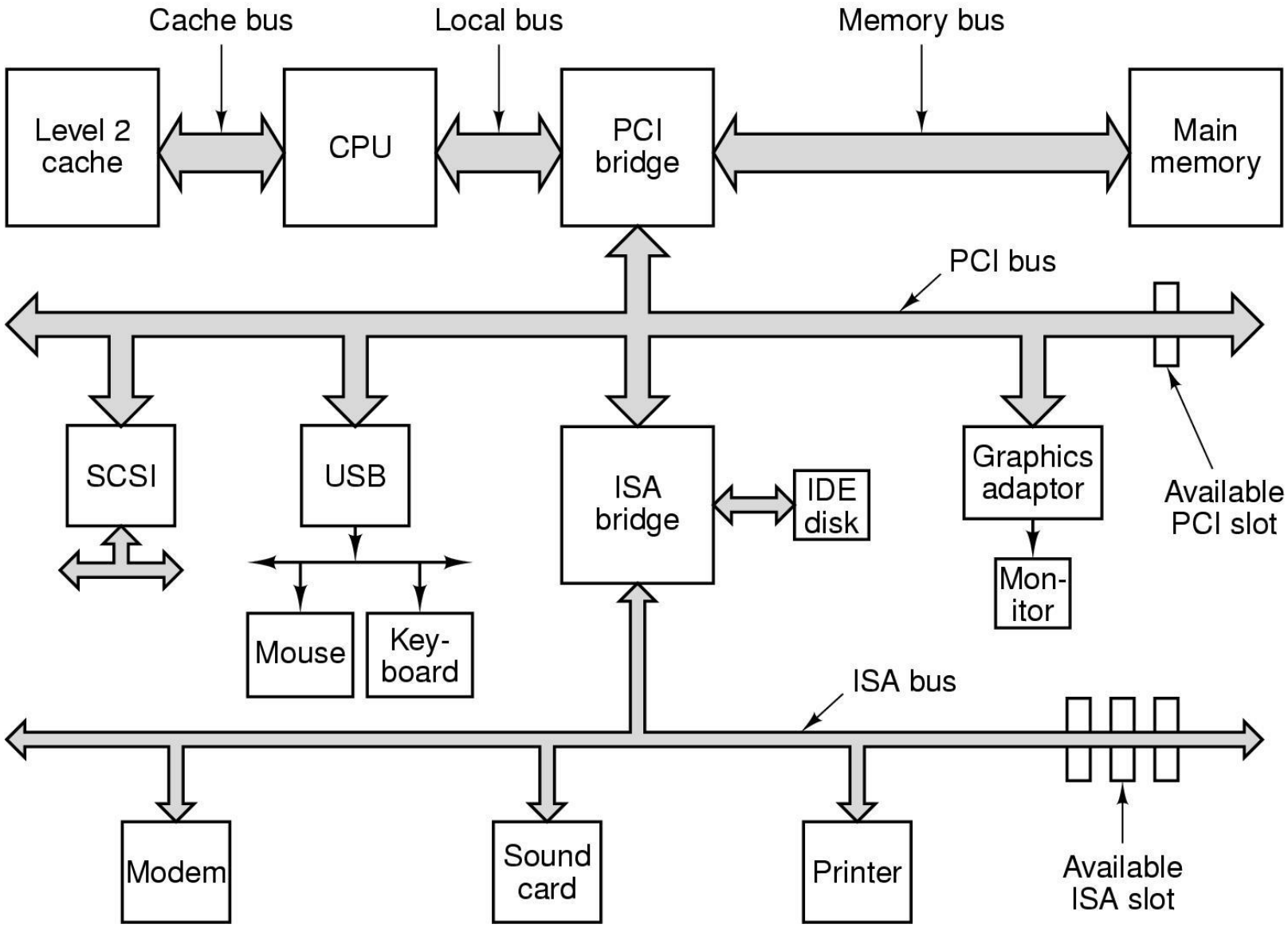
¿Qué es un Sistema Operativo (SO)?

- Según Tanenbaum
  - Un sistema operativo es …
- … una máquina extendida.
- … una capa de software que se encarga de los aspectos técnicos de una "computadora"
- … administrador de recursos.
- Provee abstracciones a los programas del usuario para utilizar los recursos.

El sistema operativo como manejador de recursos

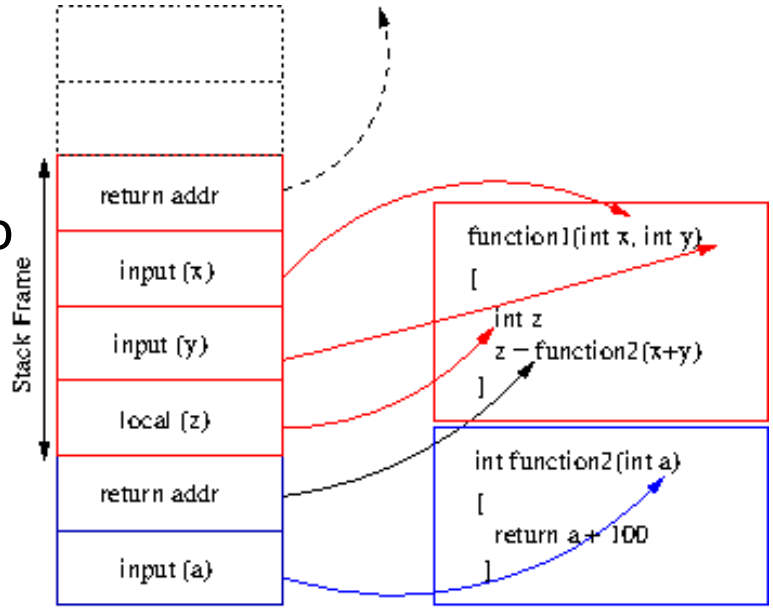


Relación del SO con el hardware

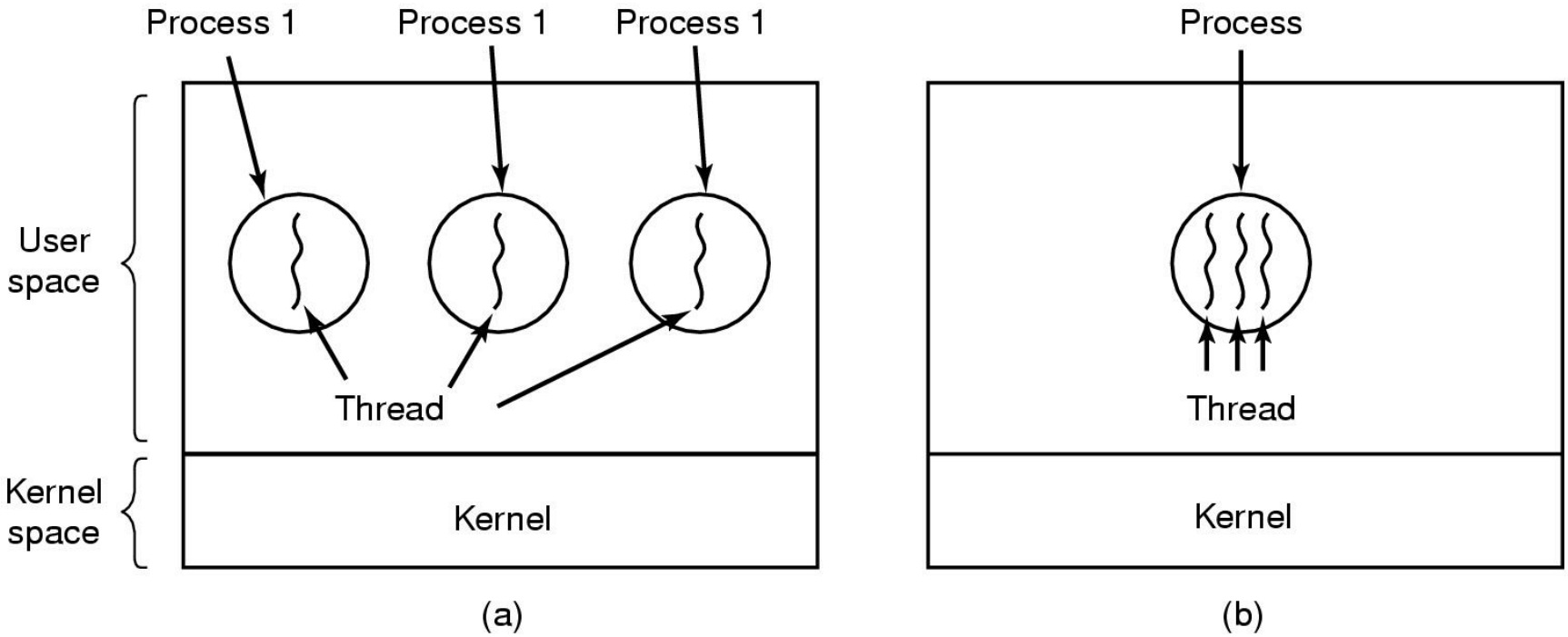


¿Qué es un proceso?

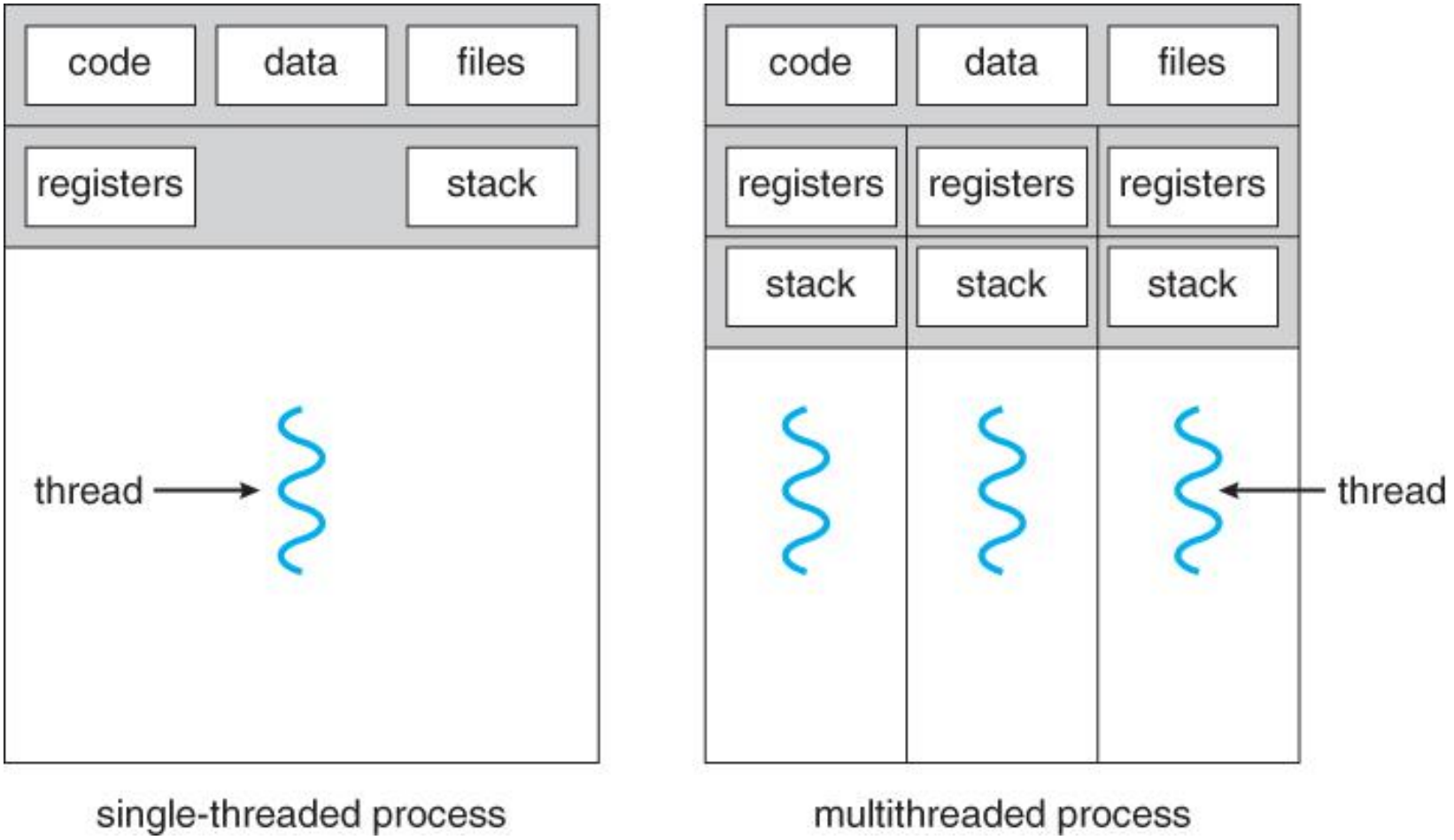
- Recipiente que contiene toda la información necesaria para ejecutar un programa
  - Espacio asociado de direcciones en memoria con programa ejecutable, datos y pila.
  - También tiene asociado un conjunto de recursos:
    - Registros de sistema (e.g., contadores)
    - Archivos abiertos
    - Alarmas
    - Lista de procesos relacionados
    - etc.



¿Qué es un hilo?



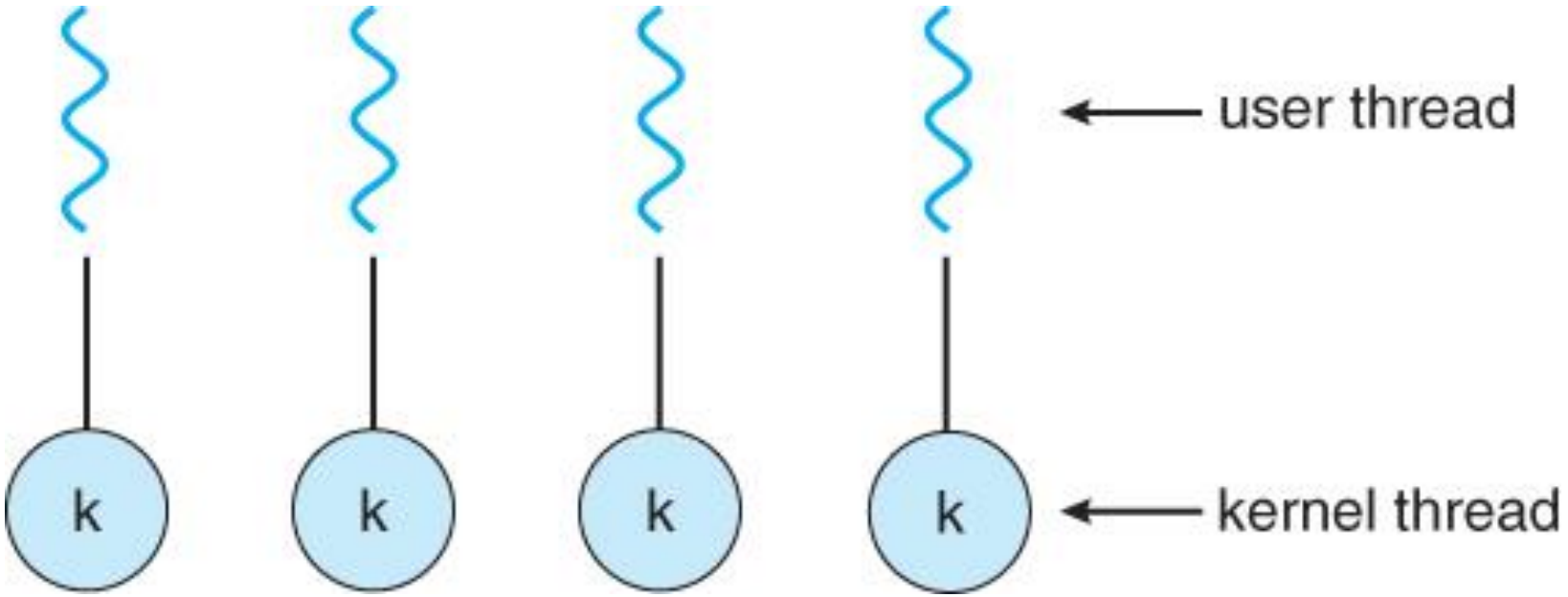
¿Qué es un hilo?



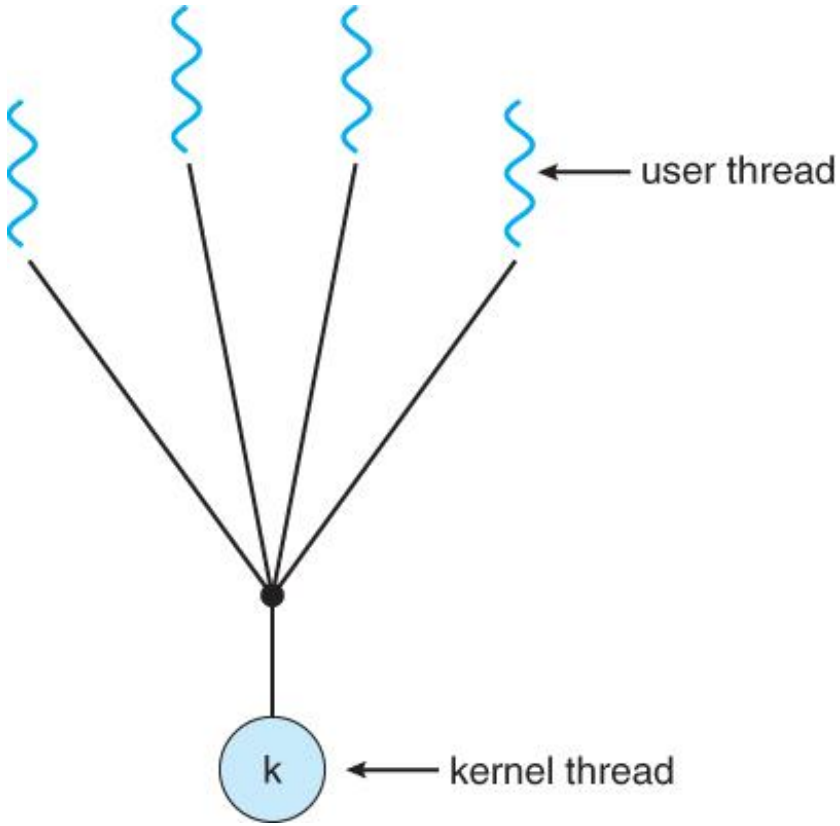
single-threaded process

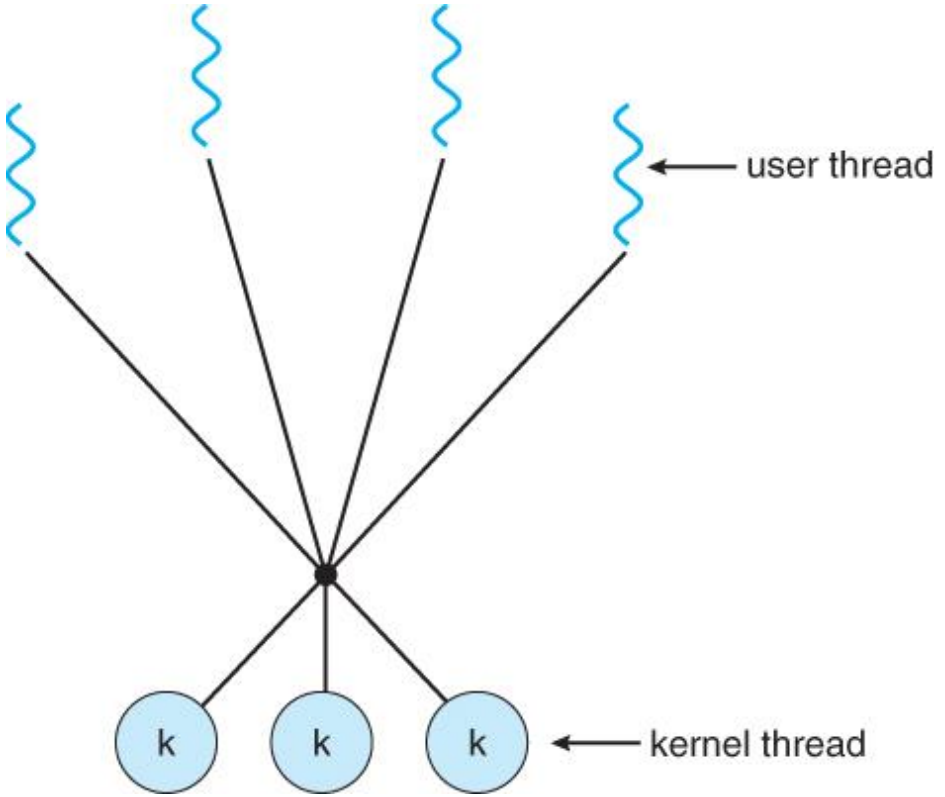
multithreaded process

Hilos de Usuario & Hilos de Kernel

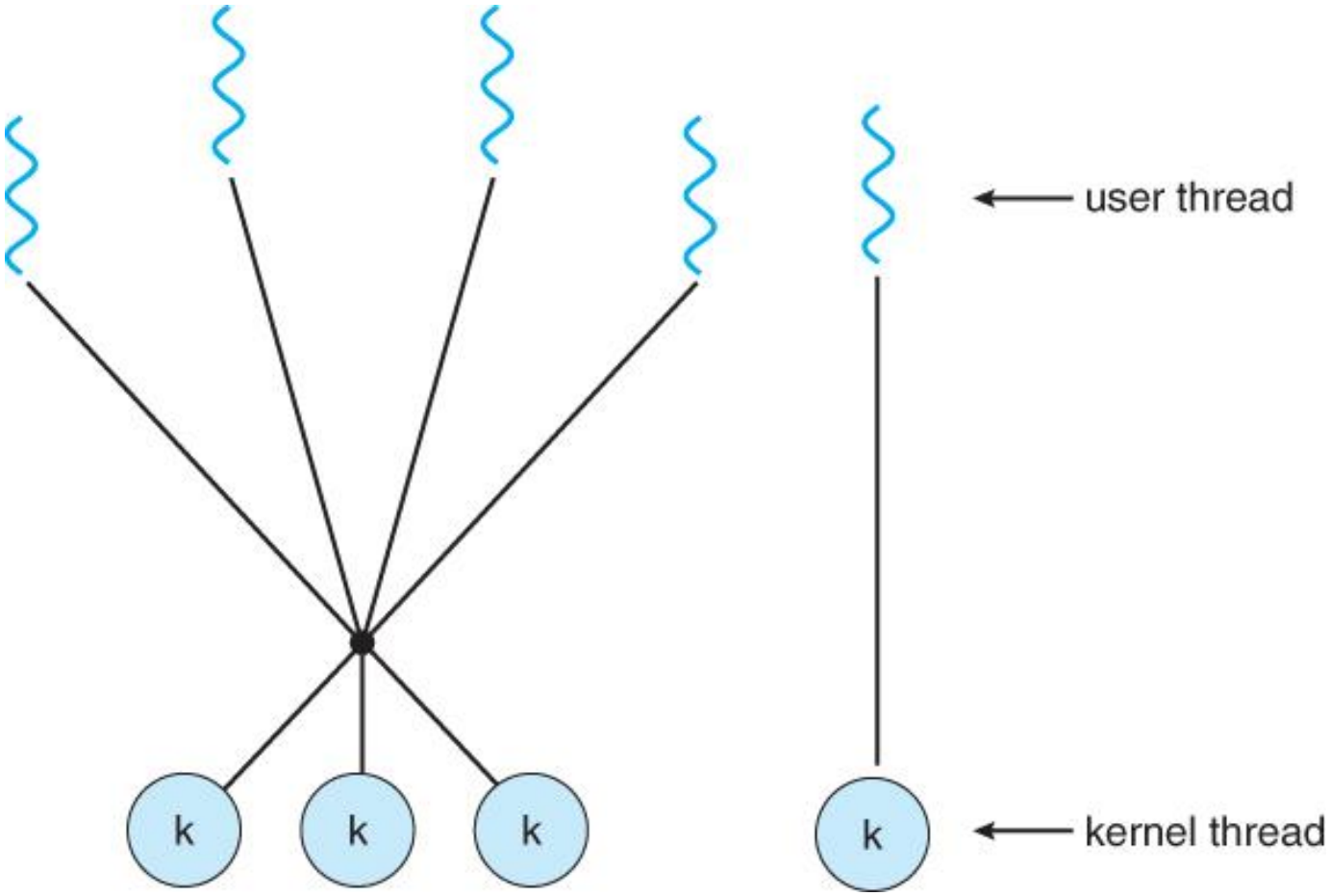


Hilos de Usuario & Hilos de Kernel



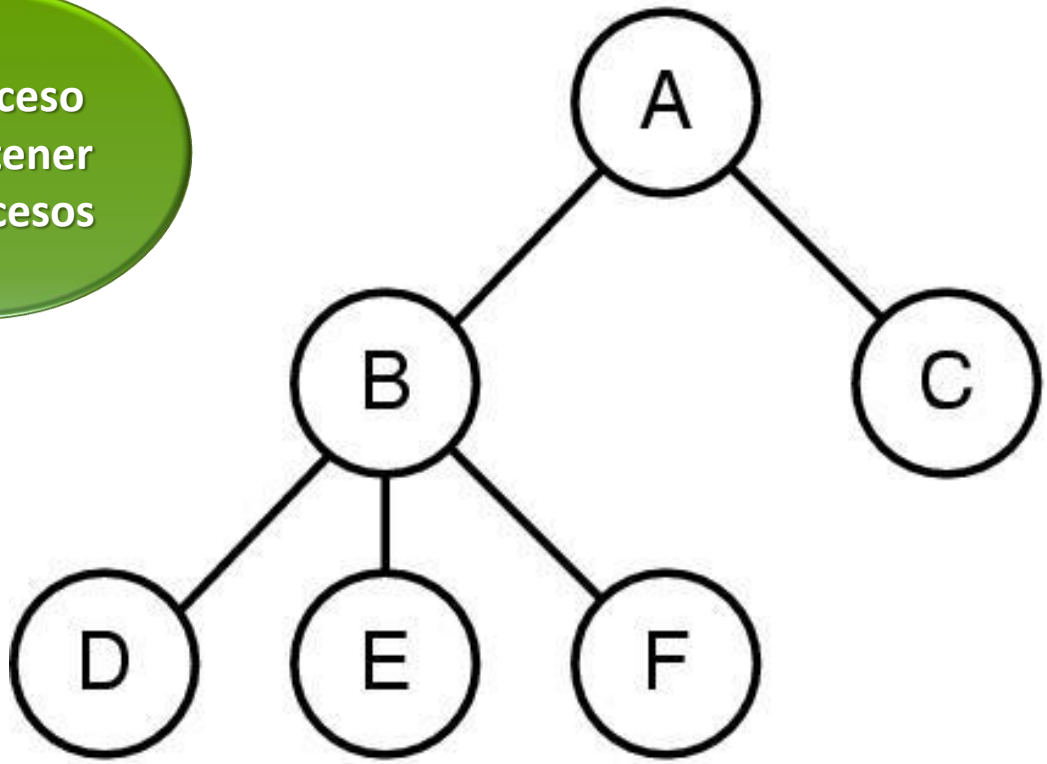


Hilos de Usuario & Hilos de Kernel

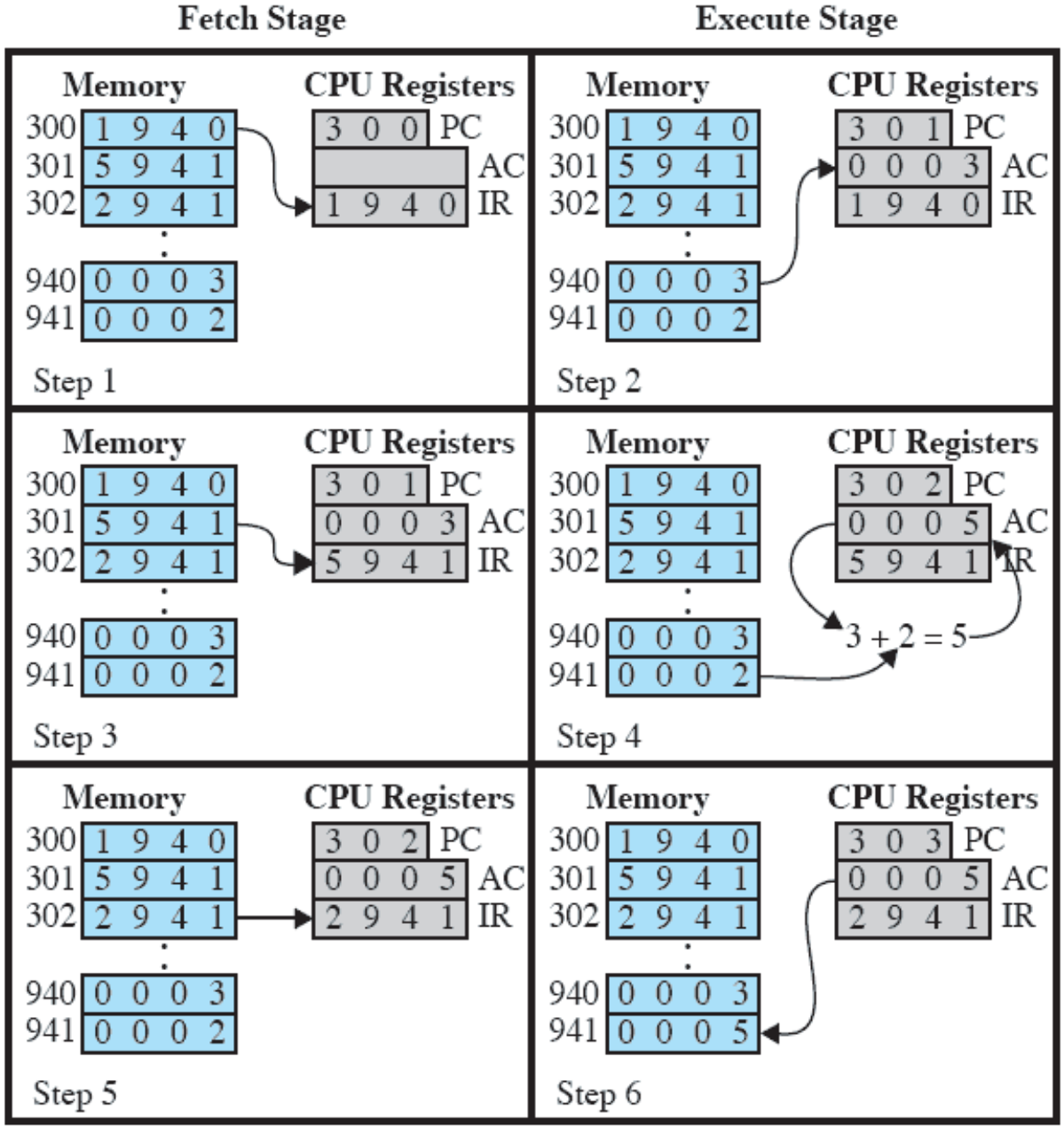


Árbol de procesos

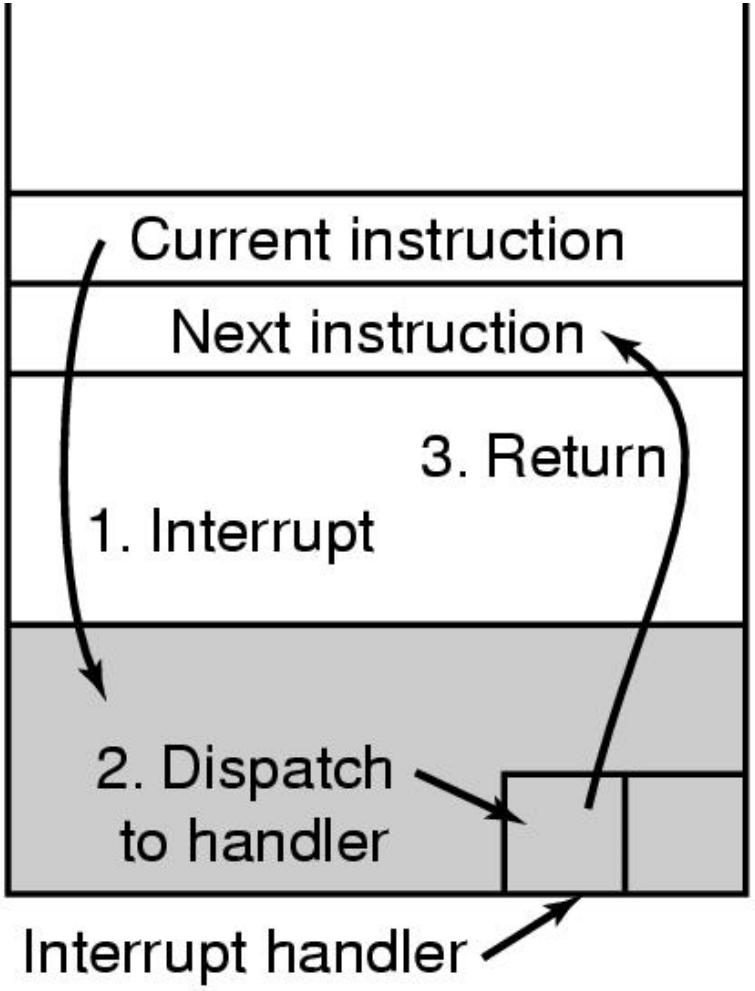
Un proceso puede tener subprocesos



Ejecución de un programa

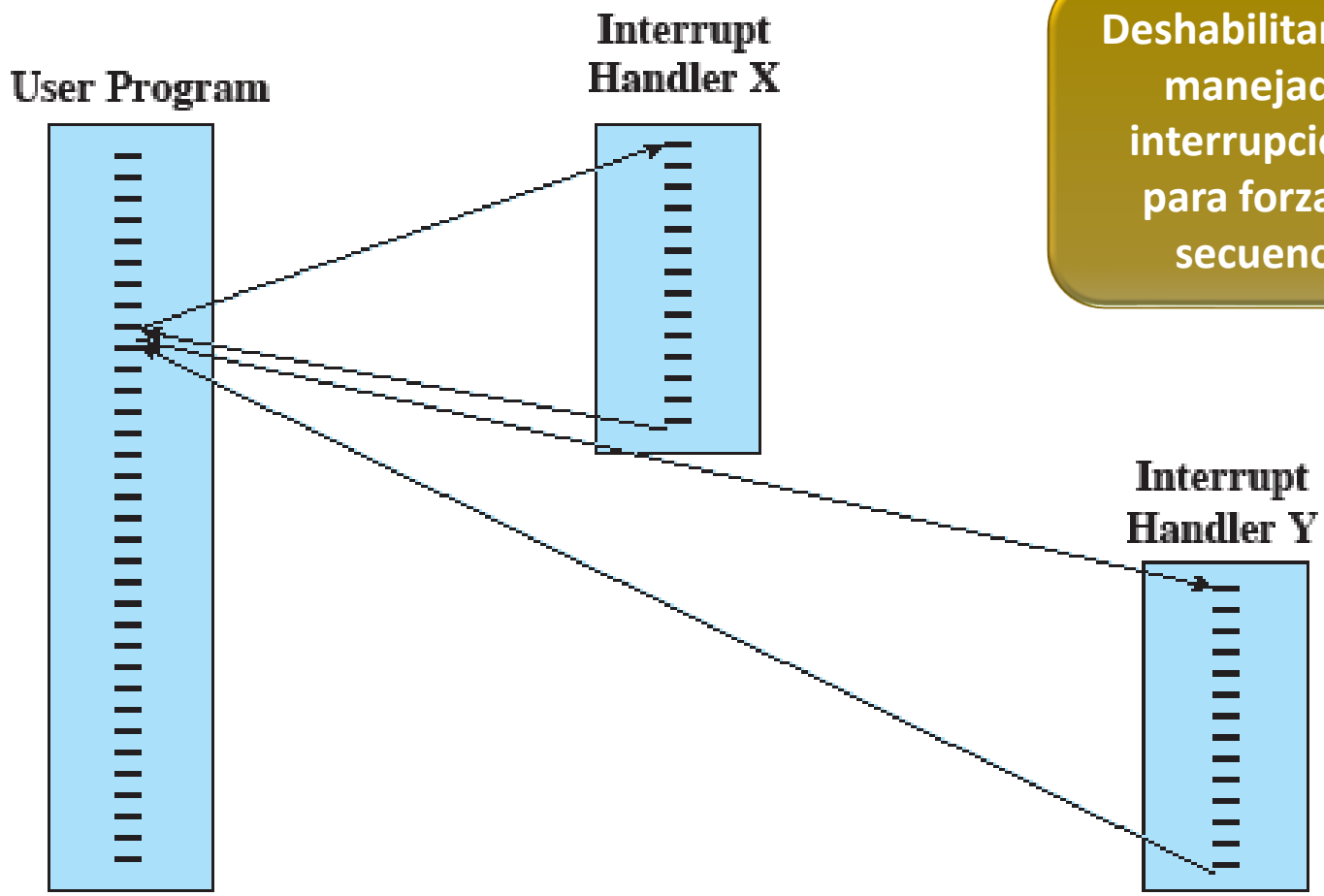


Interrupciones del sistema (o llamadas al sistema)



- Ejemplos interrupciones: creación de un archivo, lectura del teclado, despertar un hilo, etc.

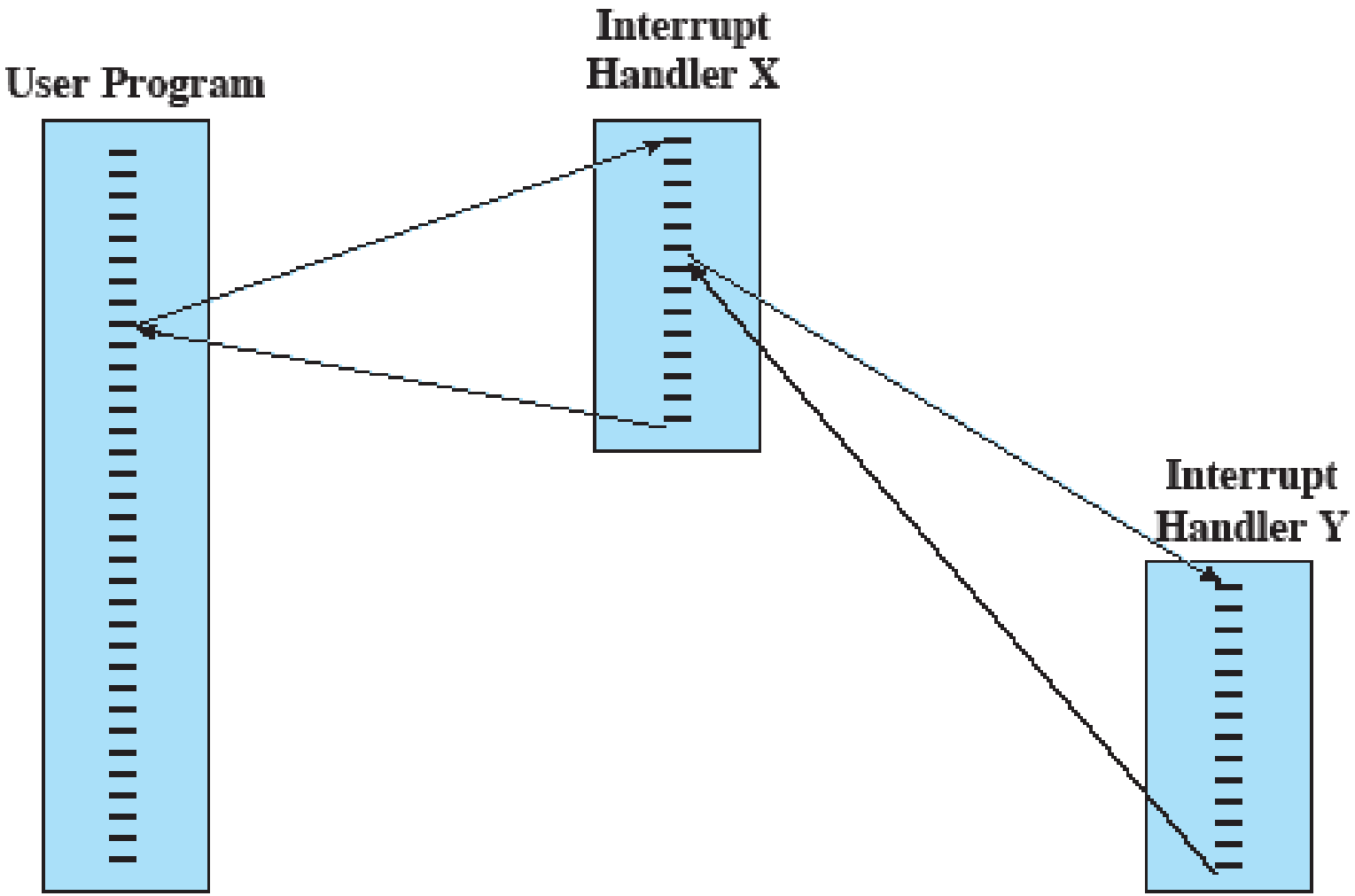
Interrupciones múltiples



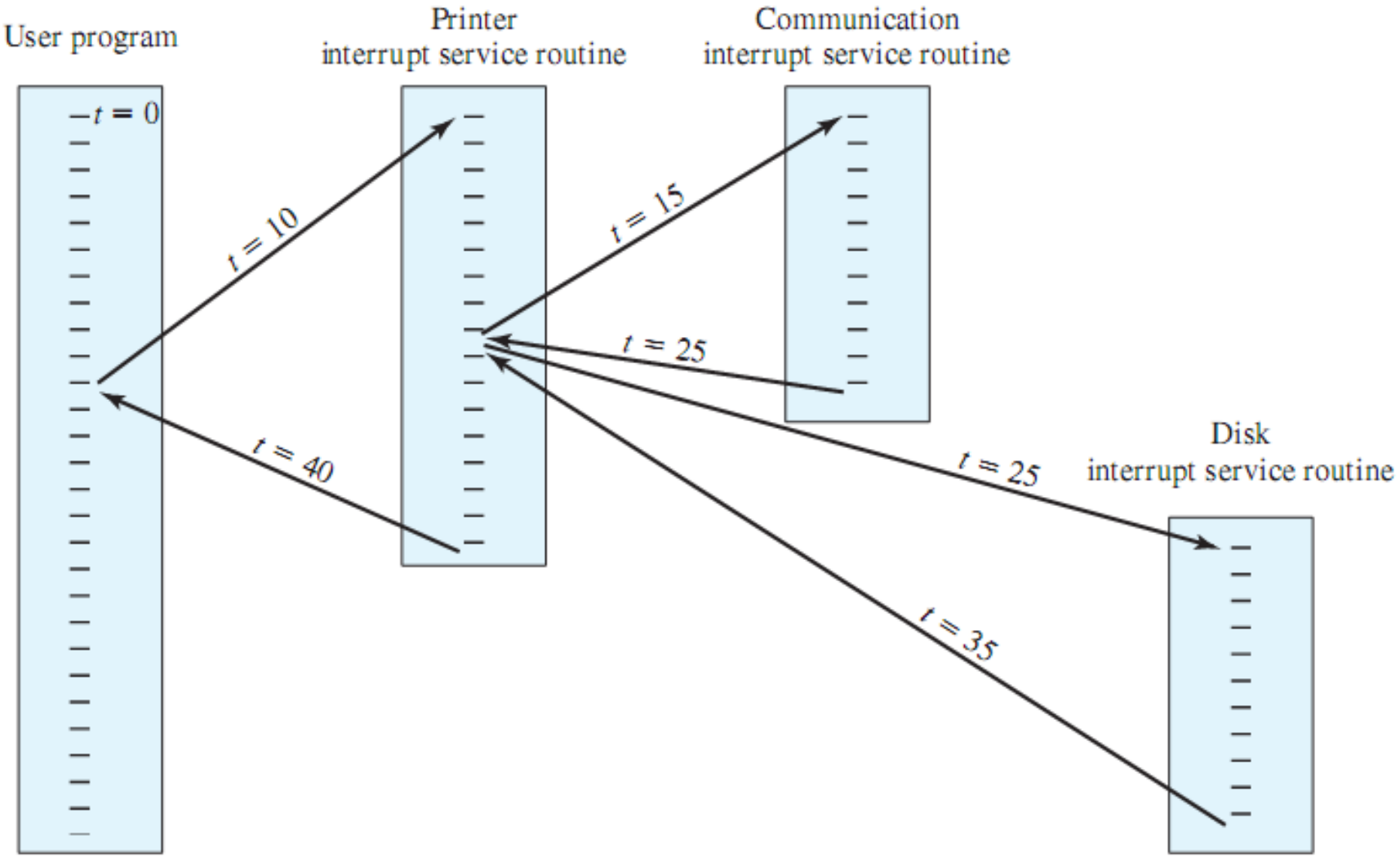
(a) Sequential interrupt processing

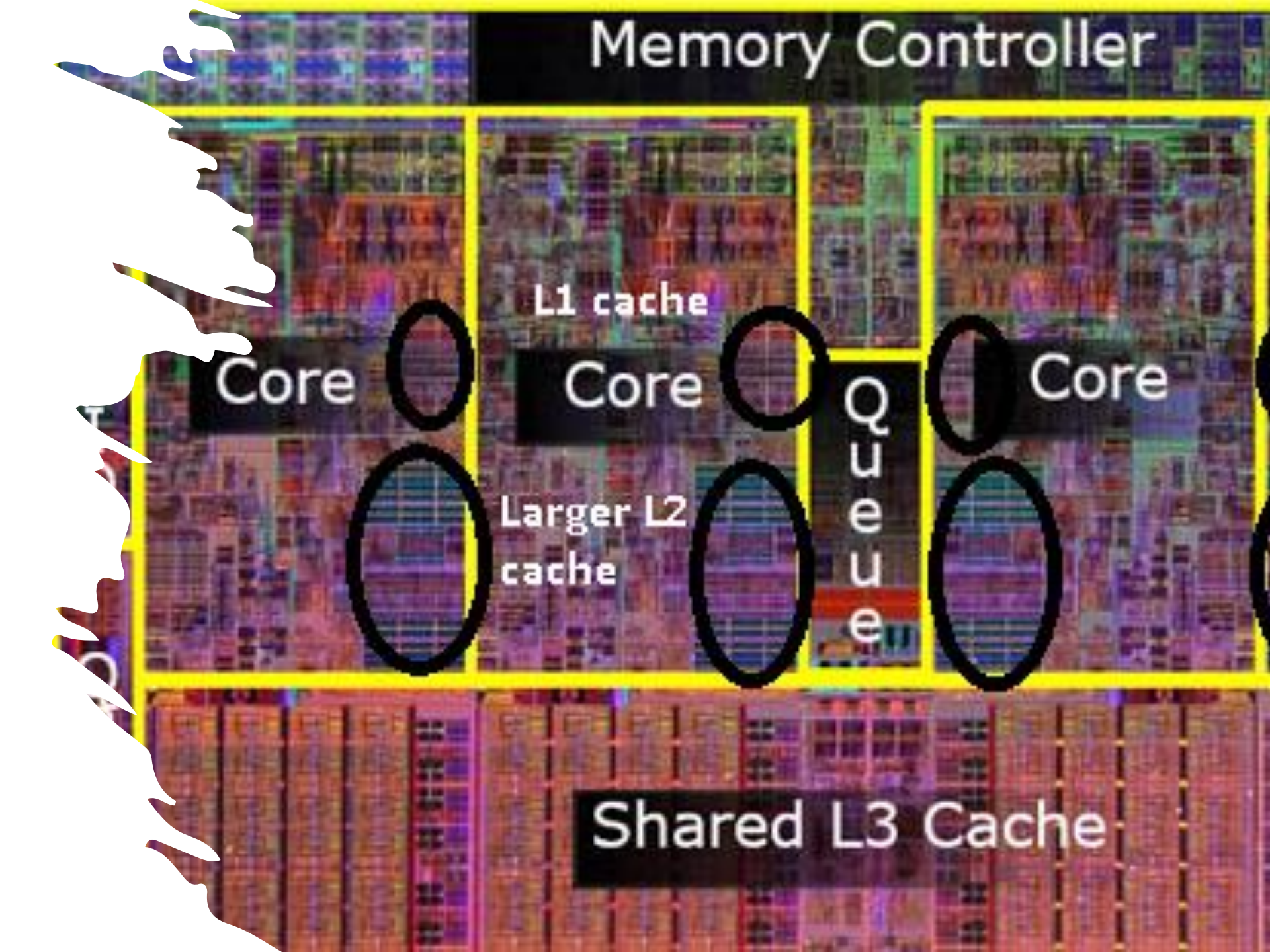
Deshabilitando el manejador interrupciones para forzar la secuencia

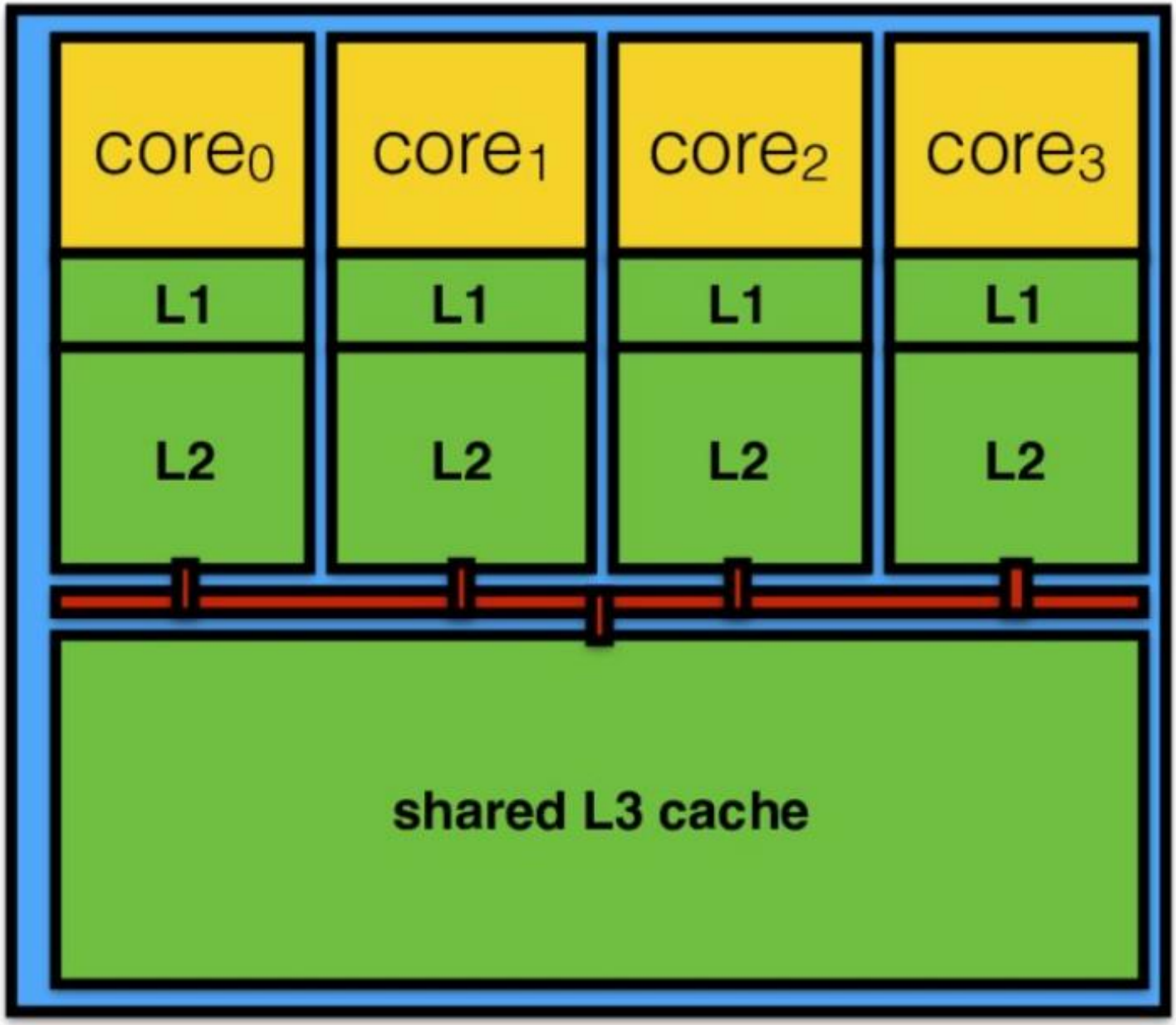
Interrupciones anidadas



Interrupciones anidadas

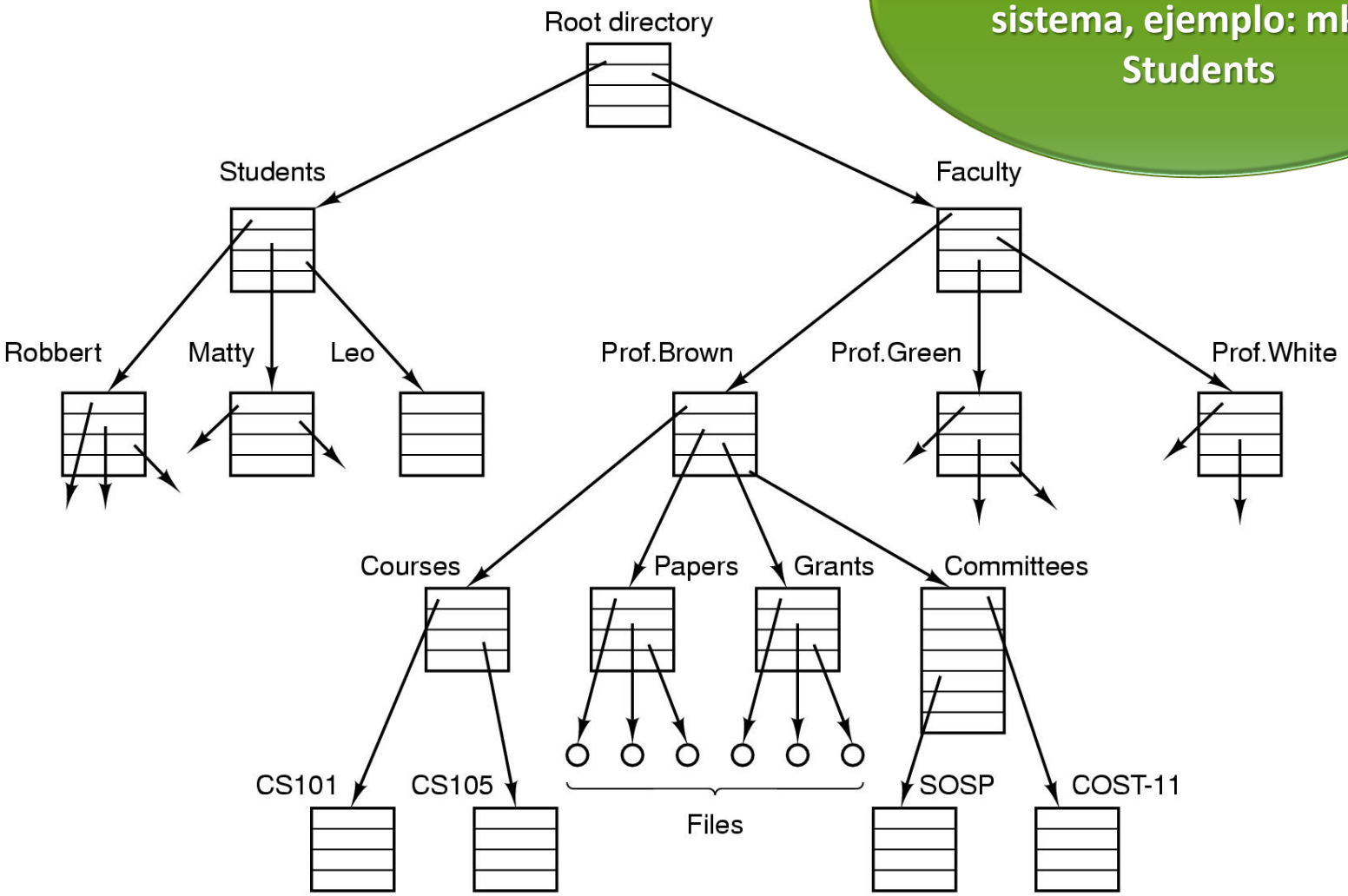



Memory Controller
L1 cache
Core
Core
Core
Larger L2 cache
Queue
Shared L3 Cache

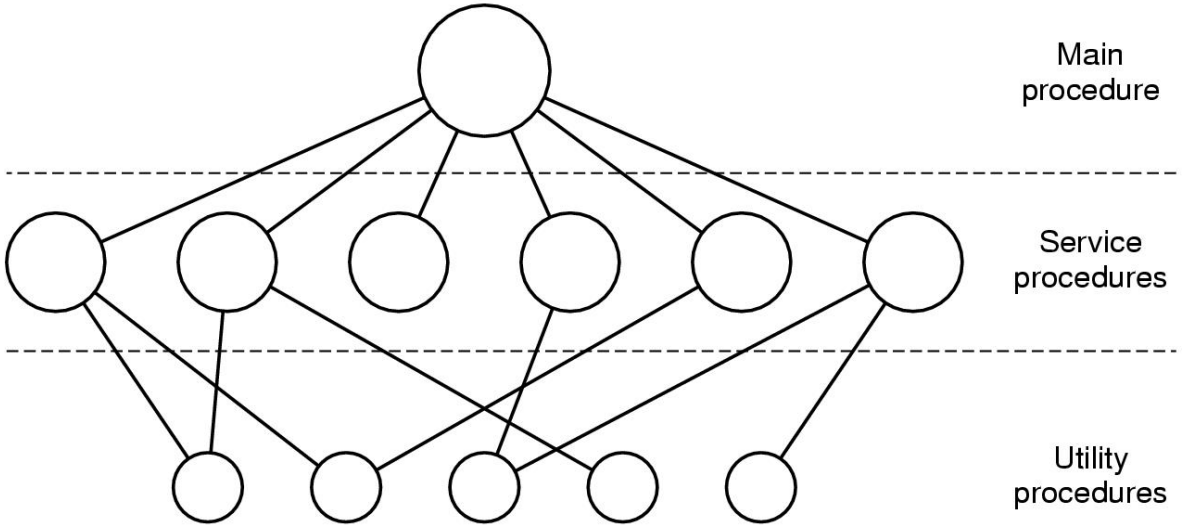

core0
core1
core2
core3
L1
L1
L1
L1
L2
L2
L2
L2
shared L3 cache

Sistema de archivos

Sistema de archivos creado vía llamadas al sistema, ejemplo: mkdir Students



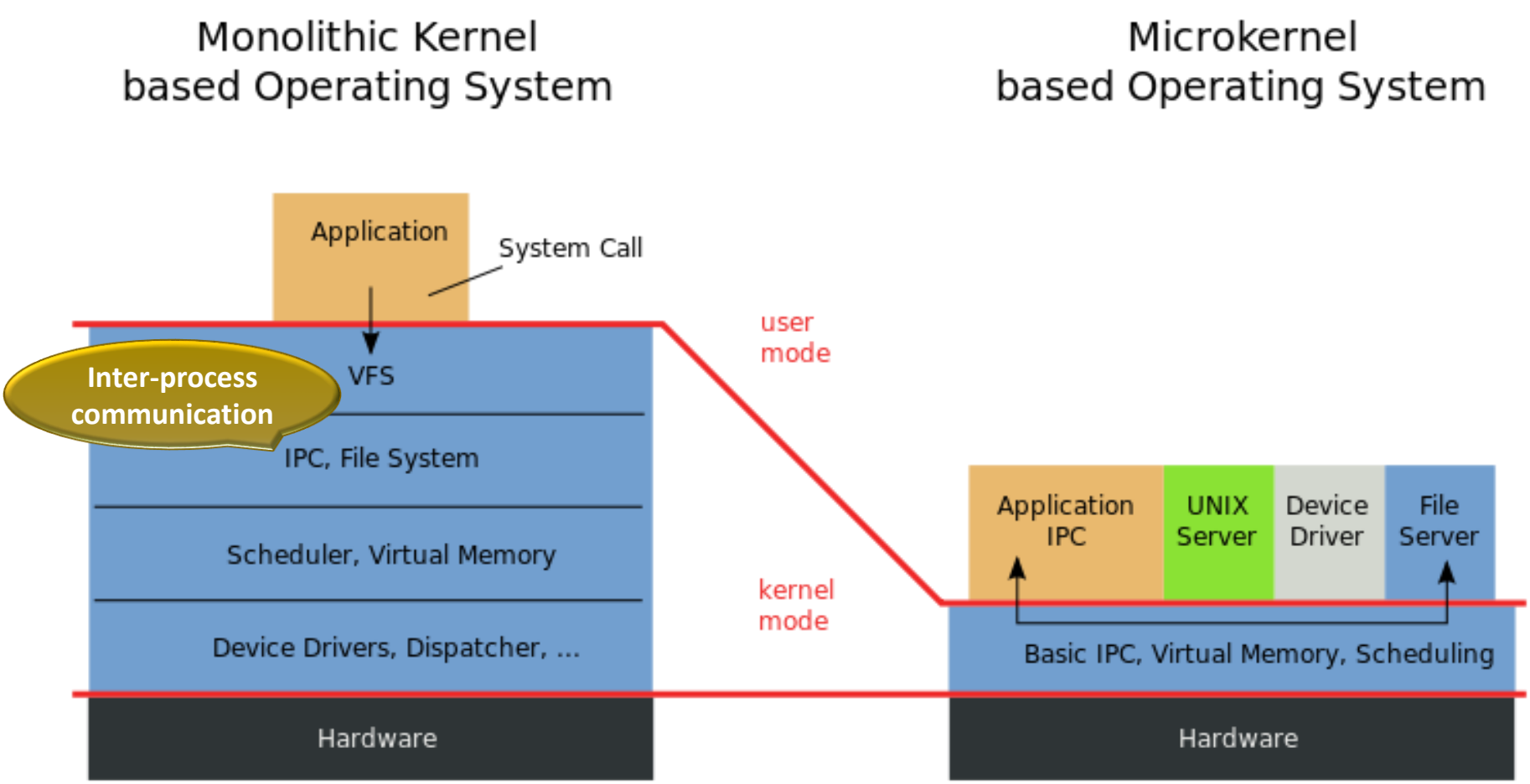
Sistema monolítico y su relación con las capas



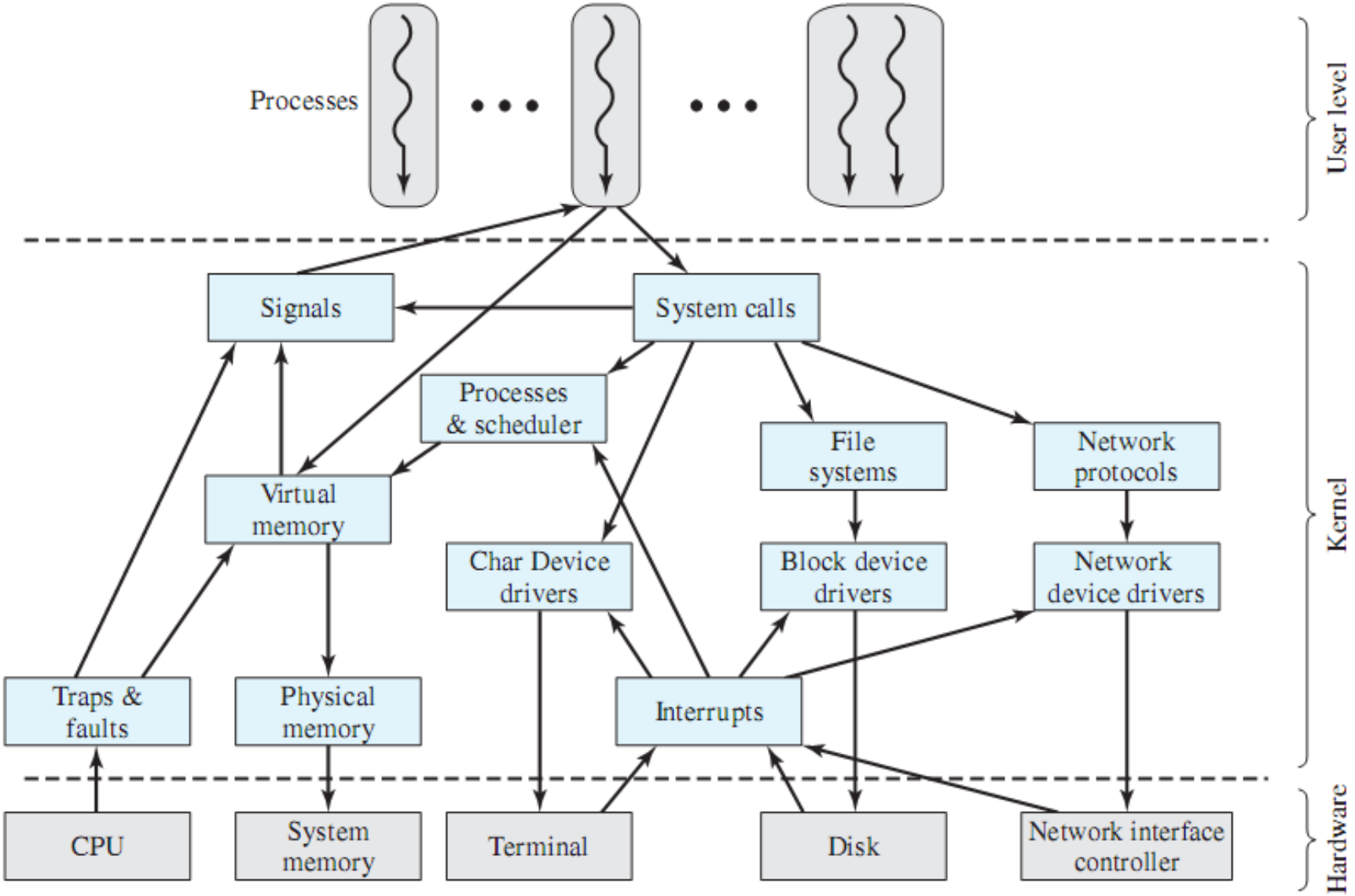
Sistema operativo THE Dijkstra

Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

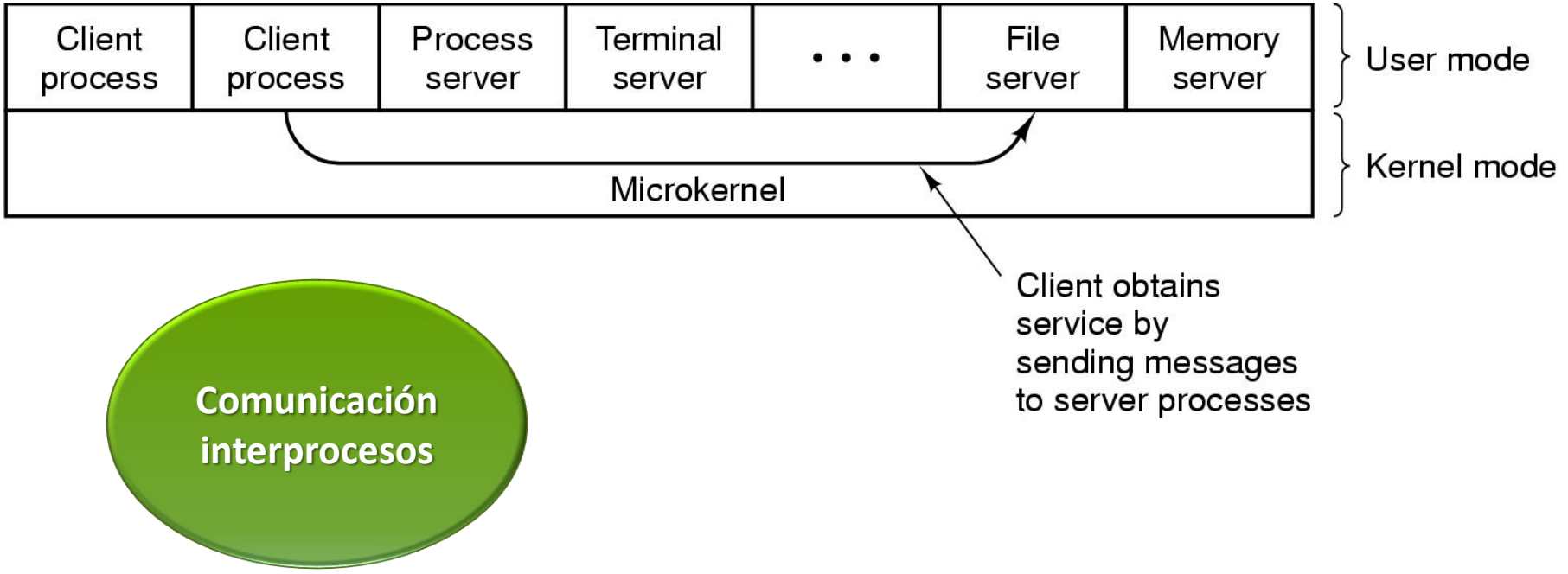
Microkernels



Componentes del Kernel de Linux

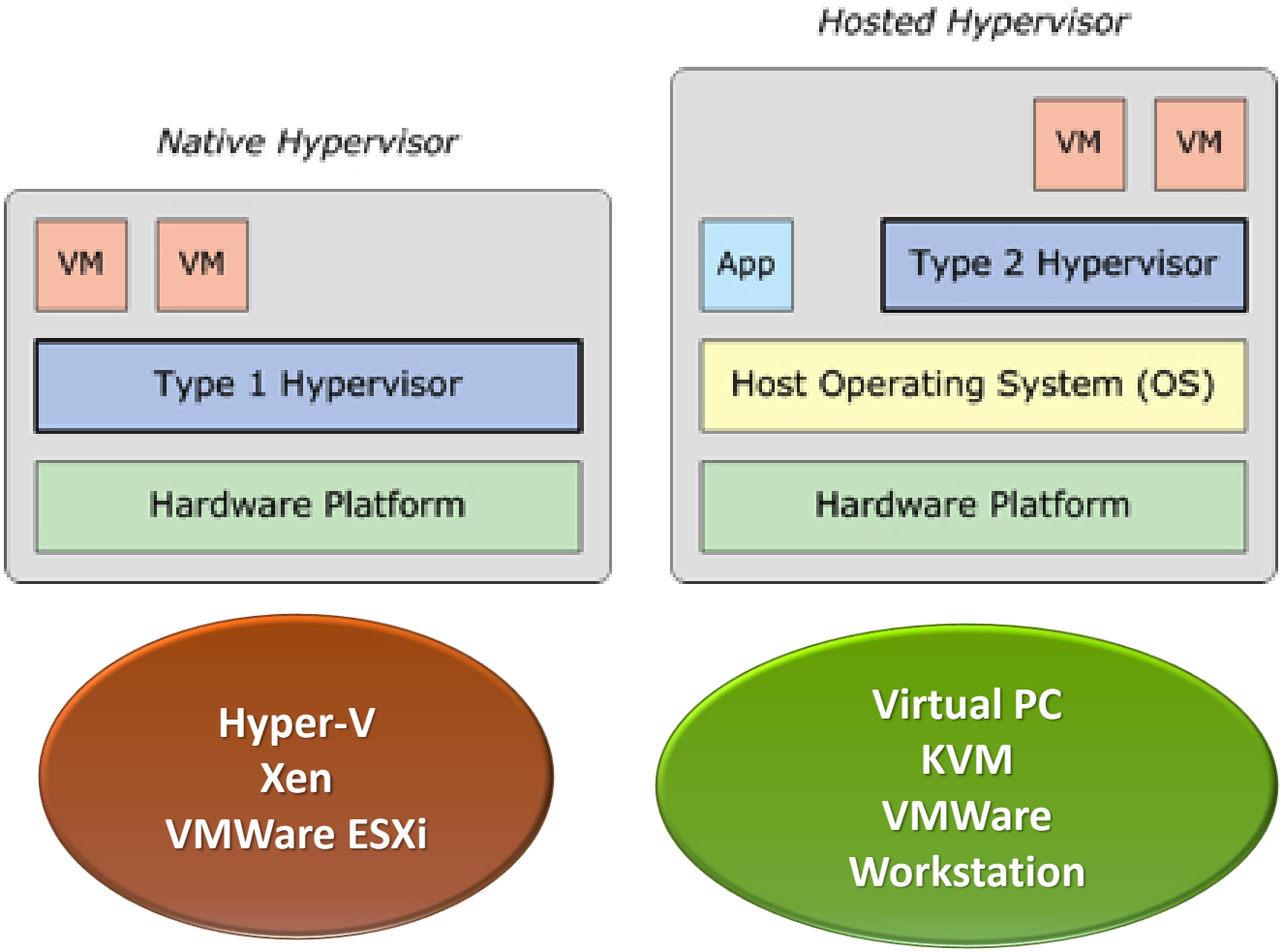


Modelo cliente servidor sobre red



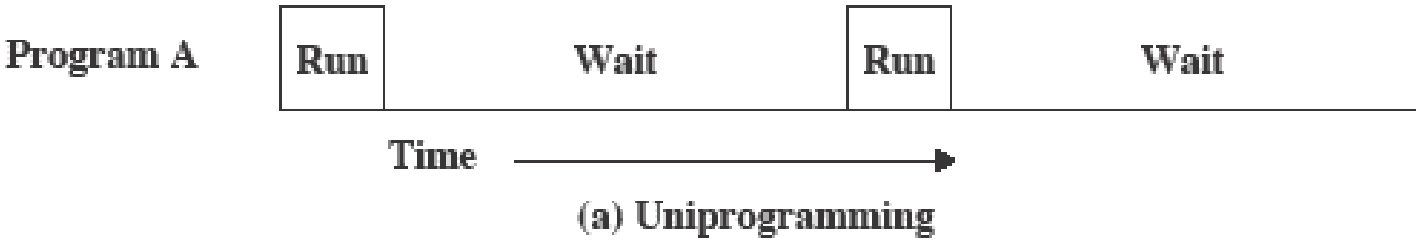
Comunicación interprocesos

Virtualización: Hypervisors



Monoprogramación

El procesador espera a que se completen llamadas I/O de un solo proceso

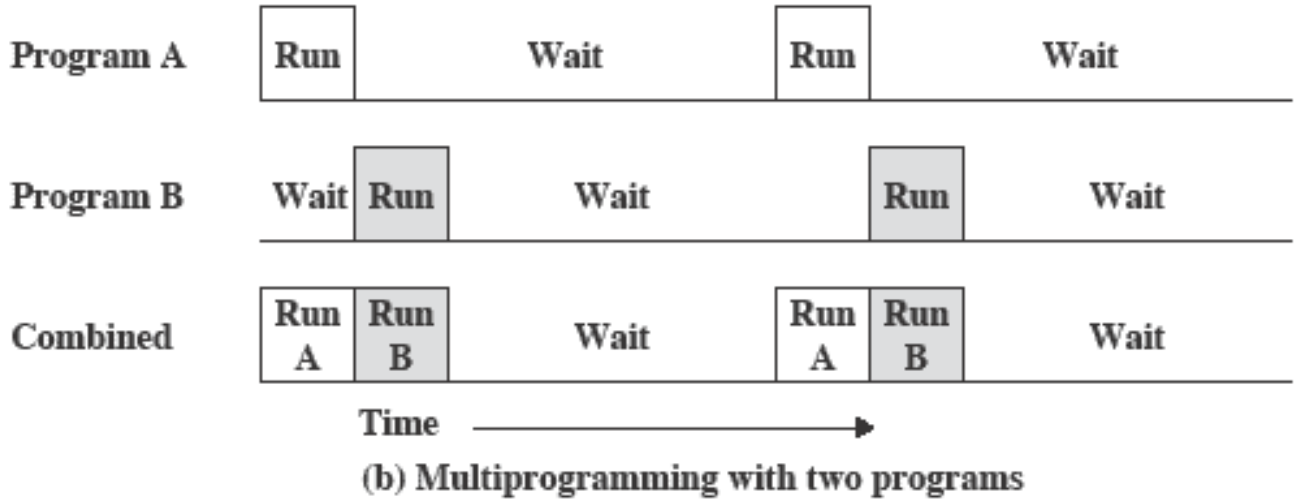


(a) Uniprogramming

Multiprogramación

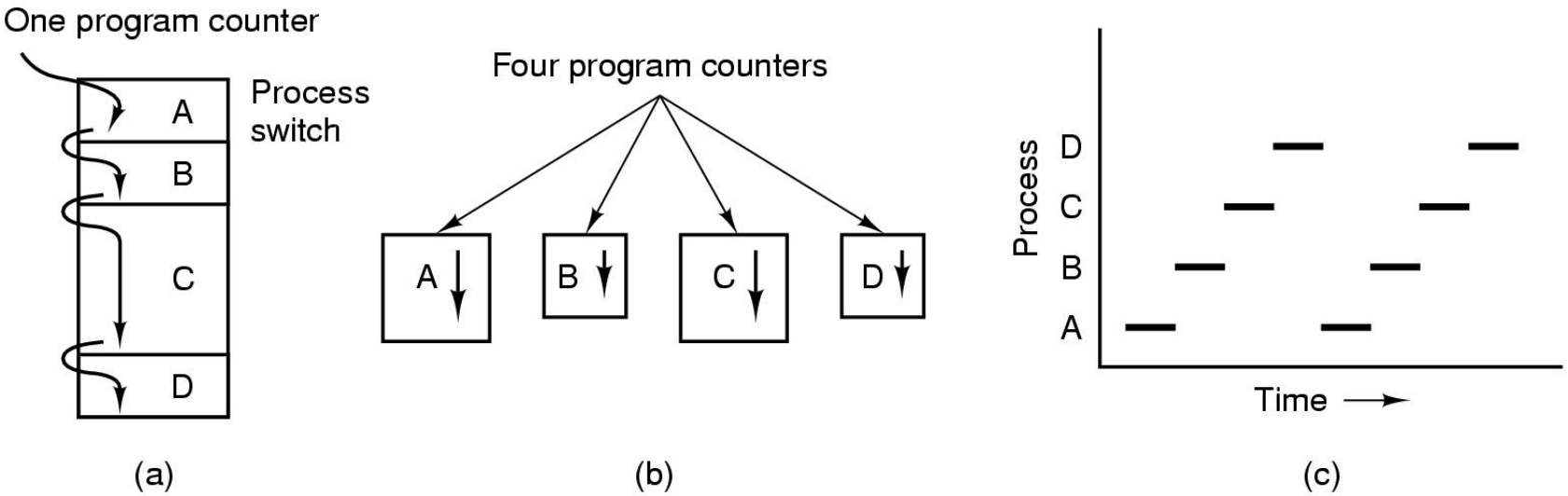
Conmutación de procesos para su ejecución en un solo procesador

Si un proceso requiere esperar a una llamadas I/O el procesador ejecuta otro proceso

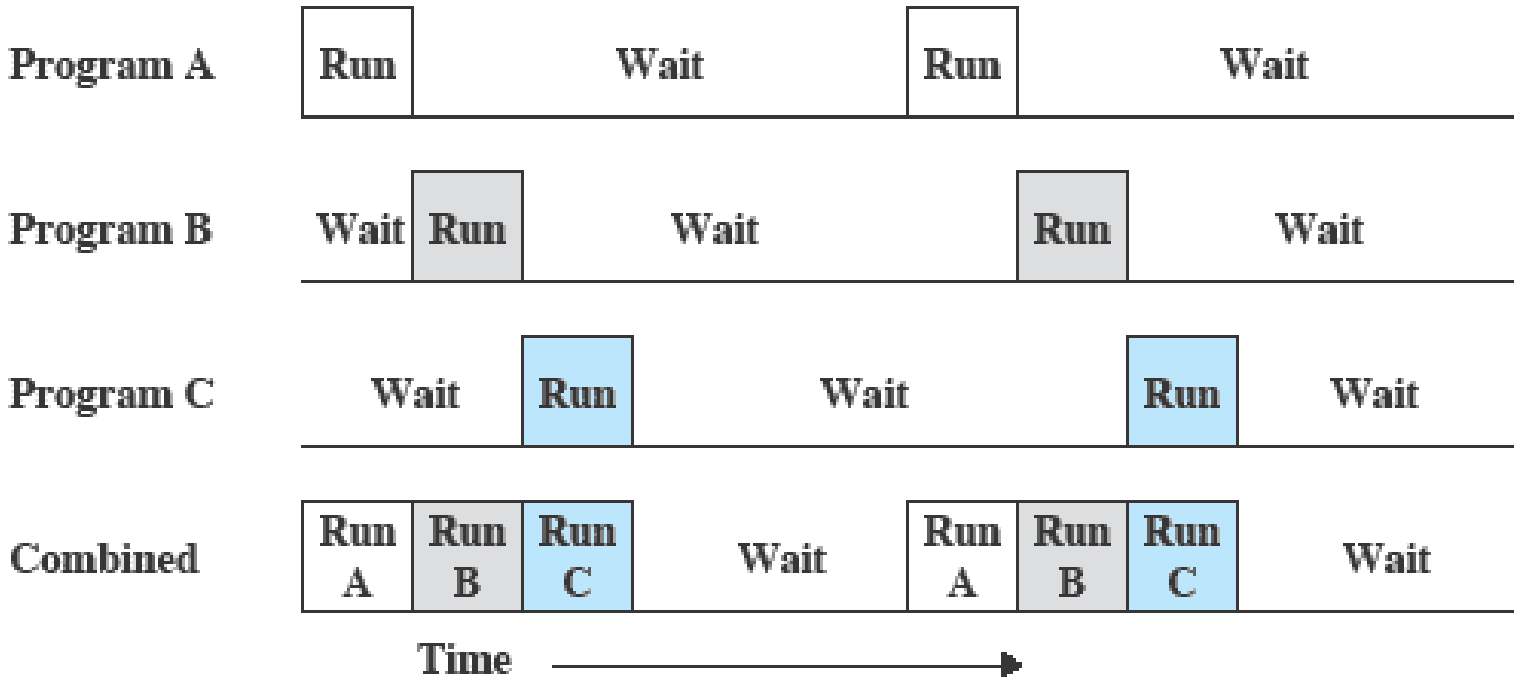


(b) Multiprogramming with two programs

Multiprogramación



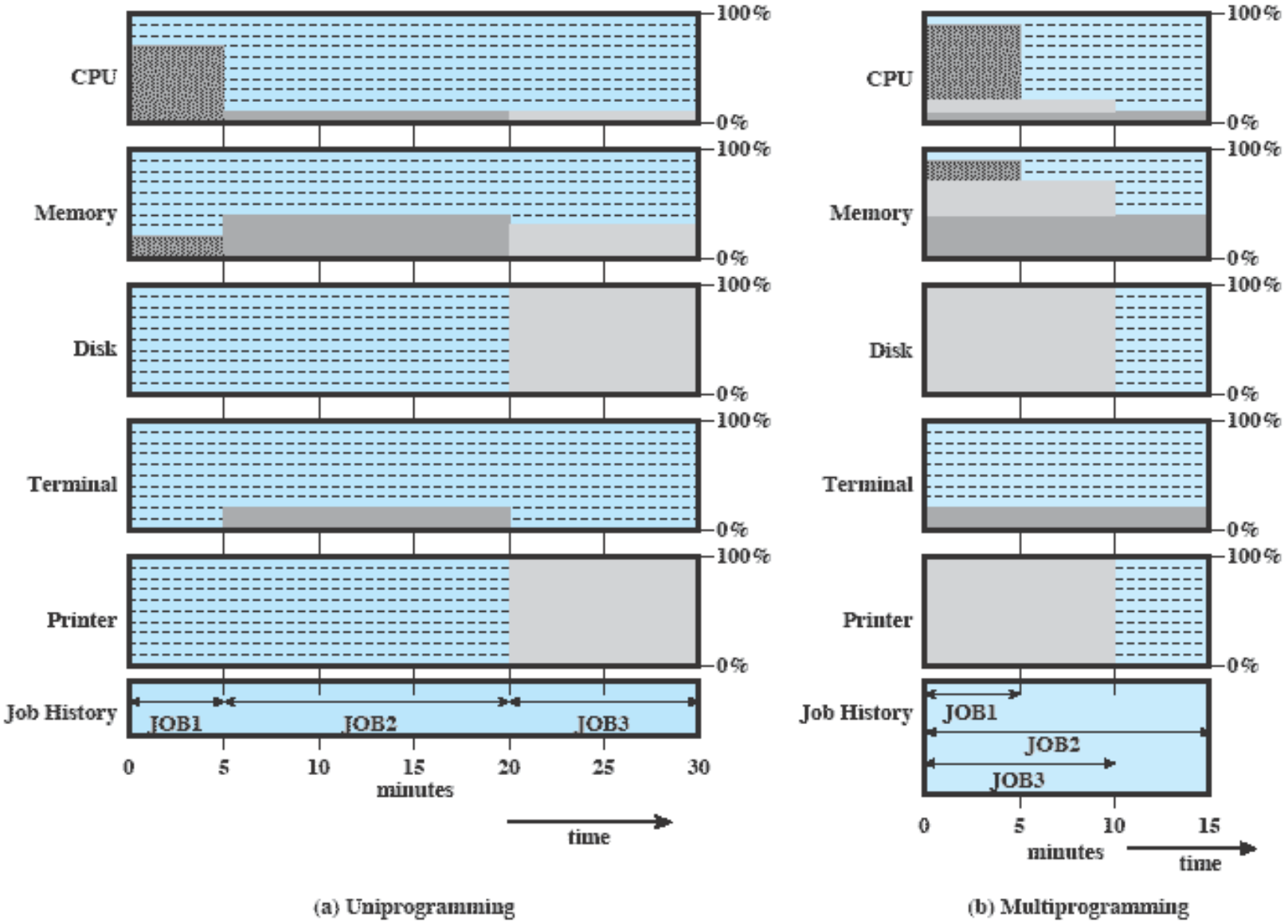
Multiprogramación



Multiprogramación

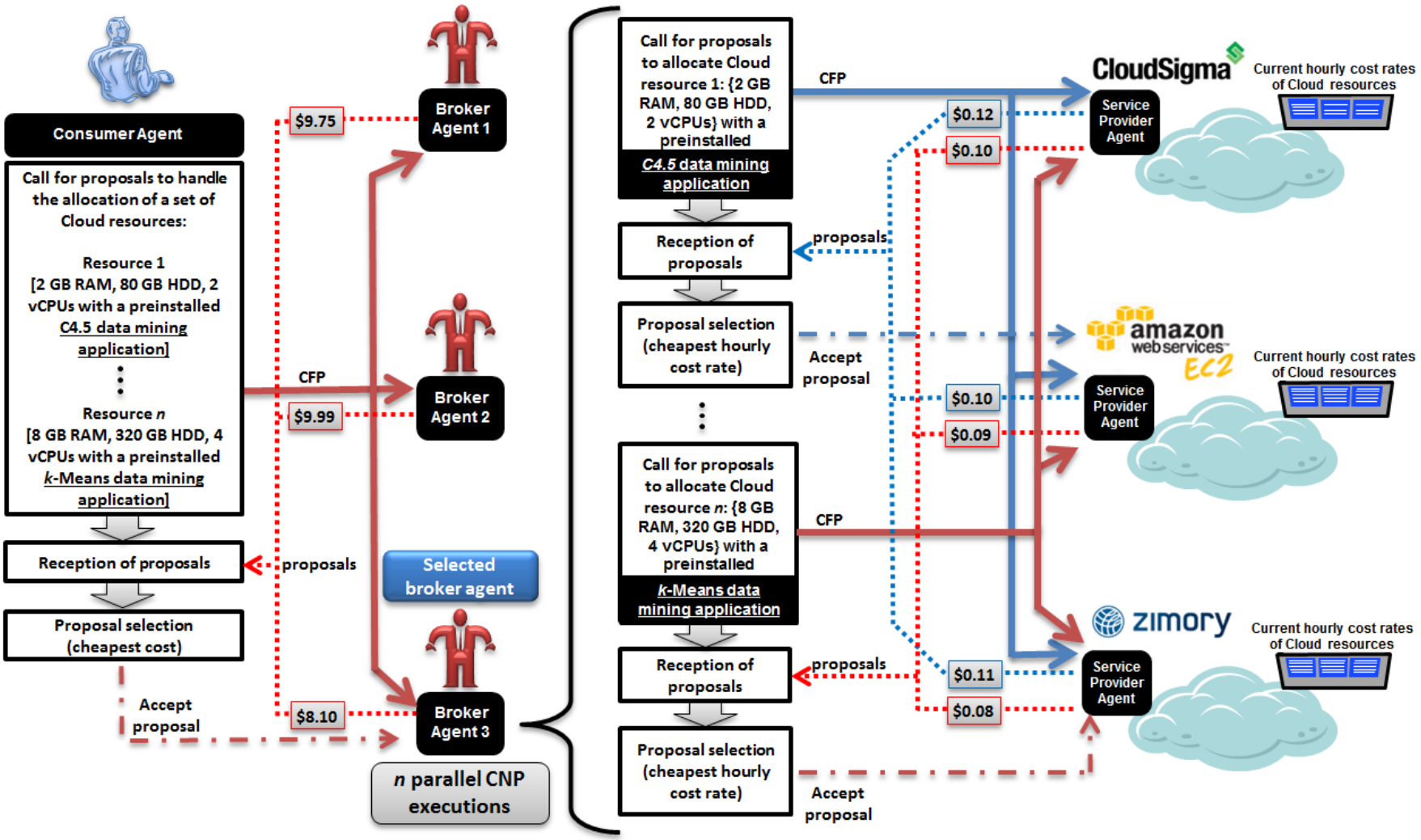
	JOB1	JOB2	JOB3
Type of job	Heavy compute	Heavy I/O	Heavy I/O
Duration	5 min	15 min	10 min
Memory required	50 M	100 M	75 M
Need disk?	No	No	Yes
Need terminal?	No	Yes	No
Need printer?	No	No	Yes

Mono VS Multiprogramación



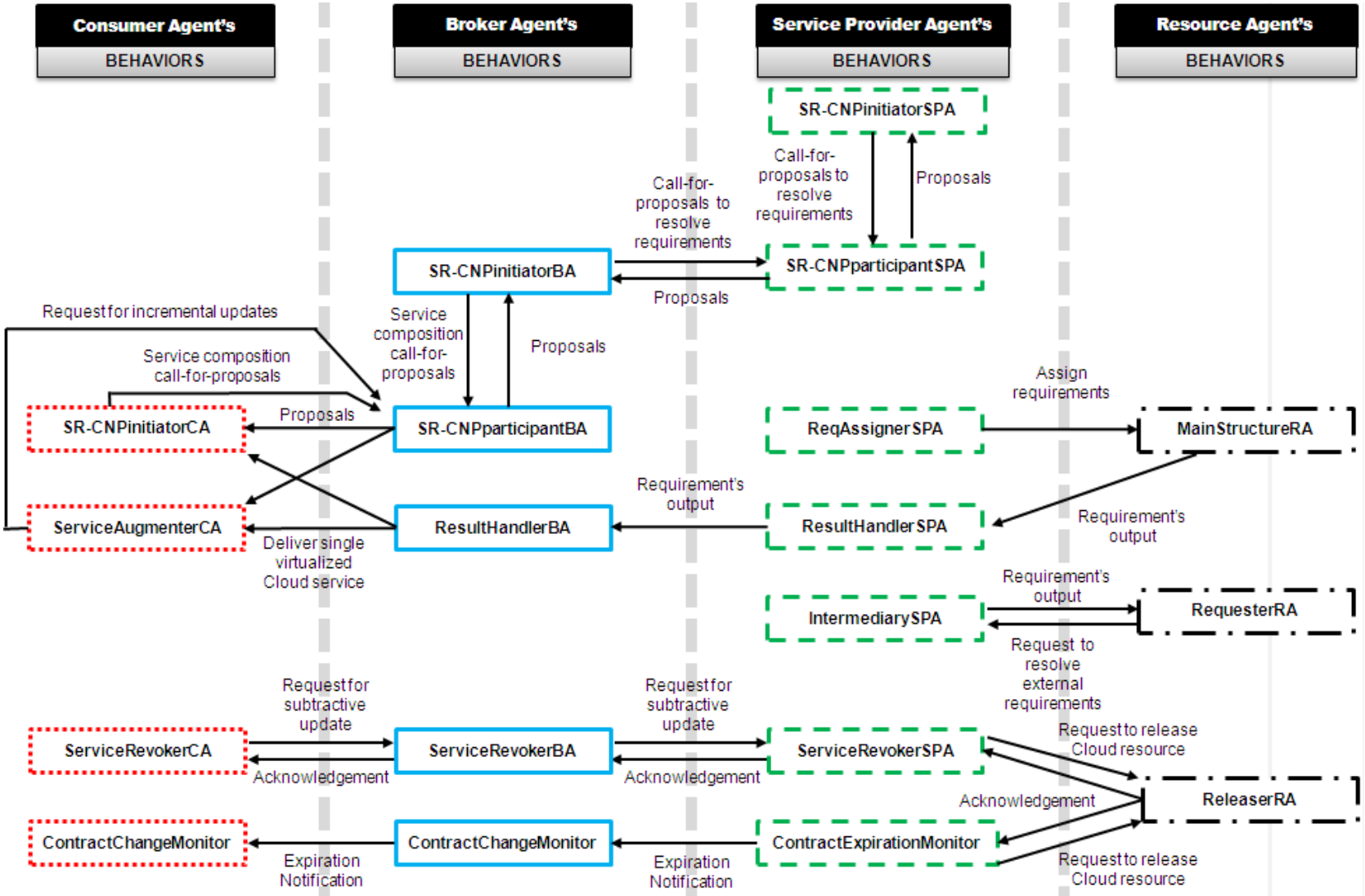
Un ejemplo de procesos con múltiples hilos:

Reserva de Recursos Cloud

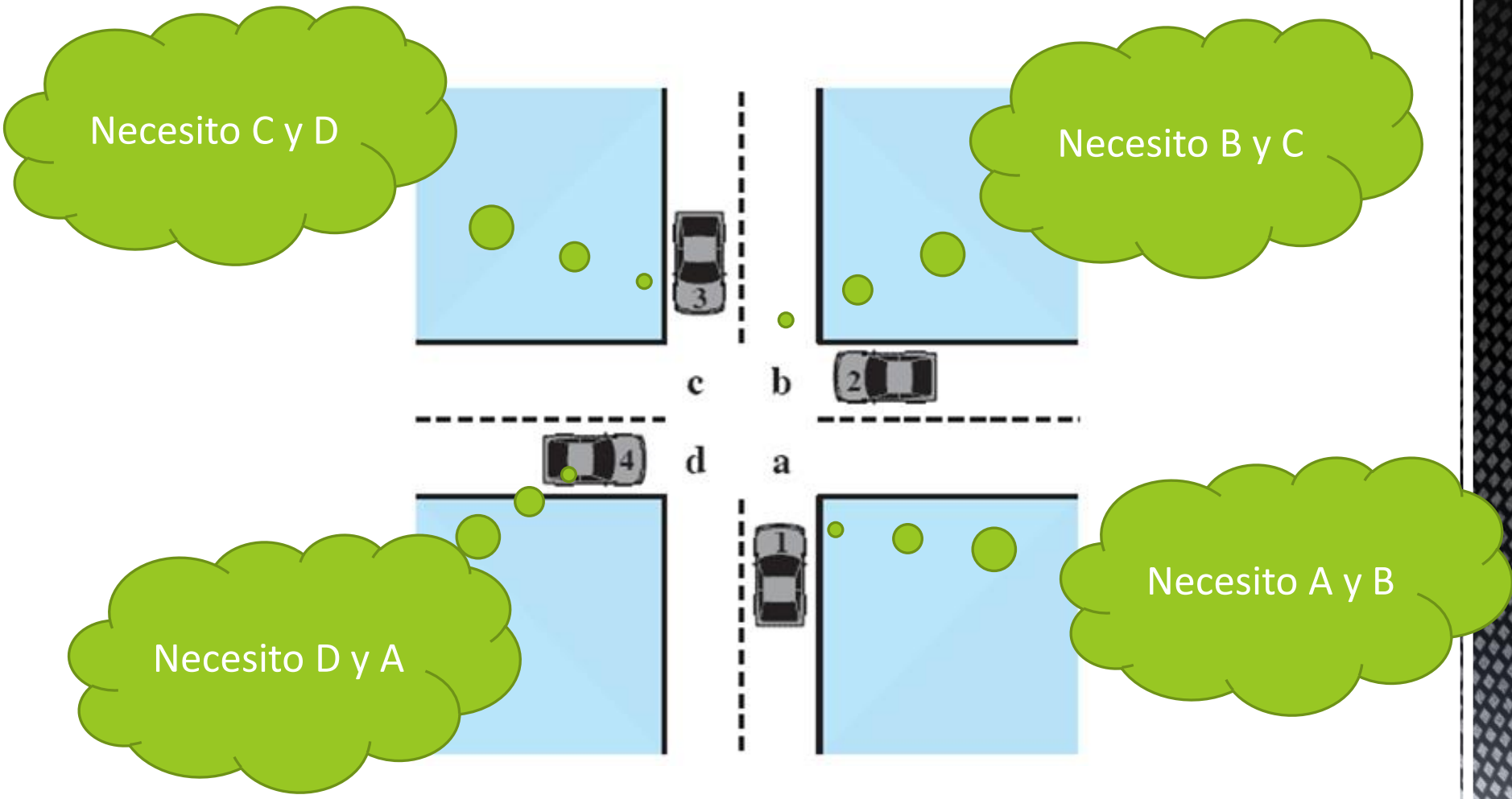


Otro ejemplo de procesos con múltiples hilos:

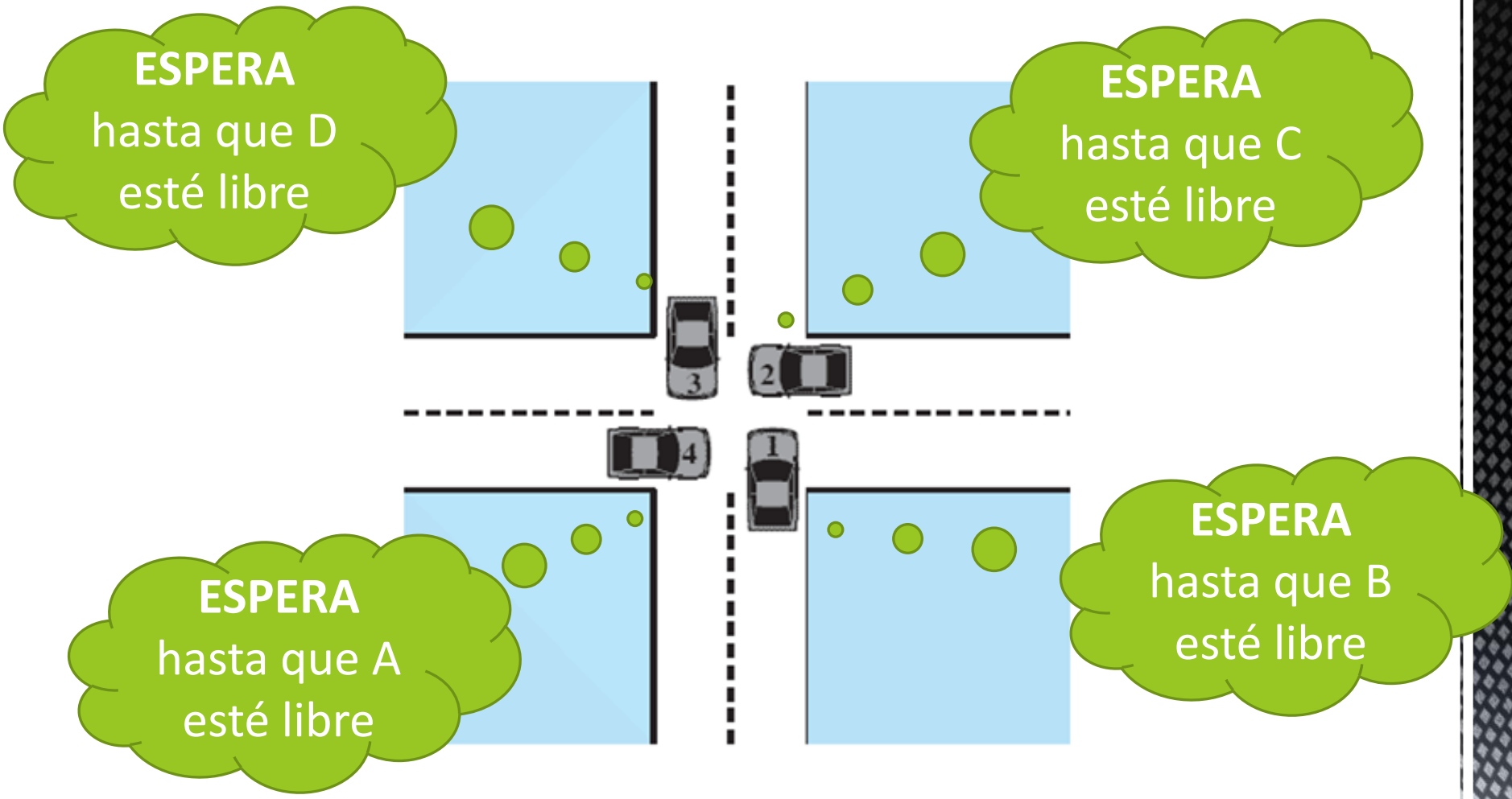
Composición de Servicios Cloud



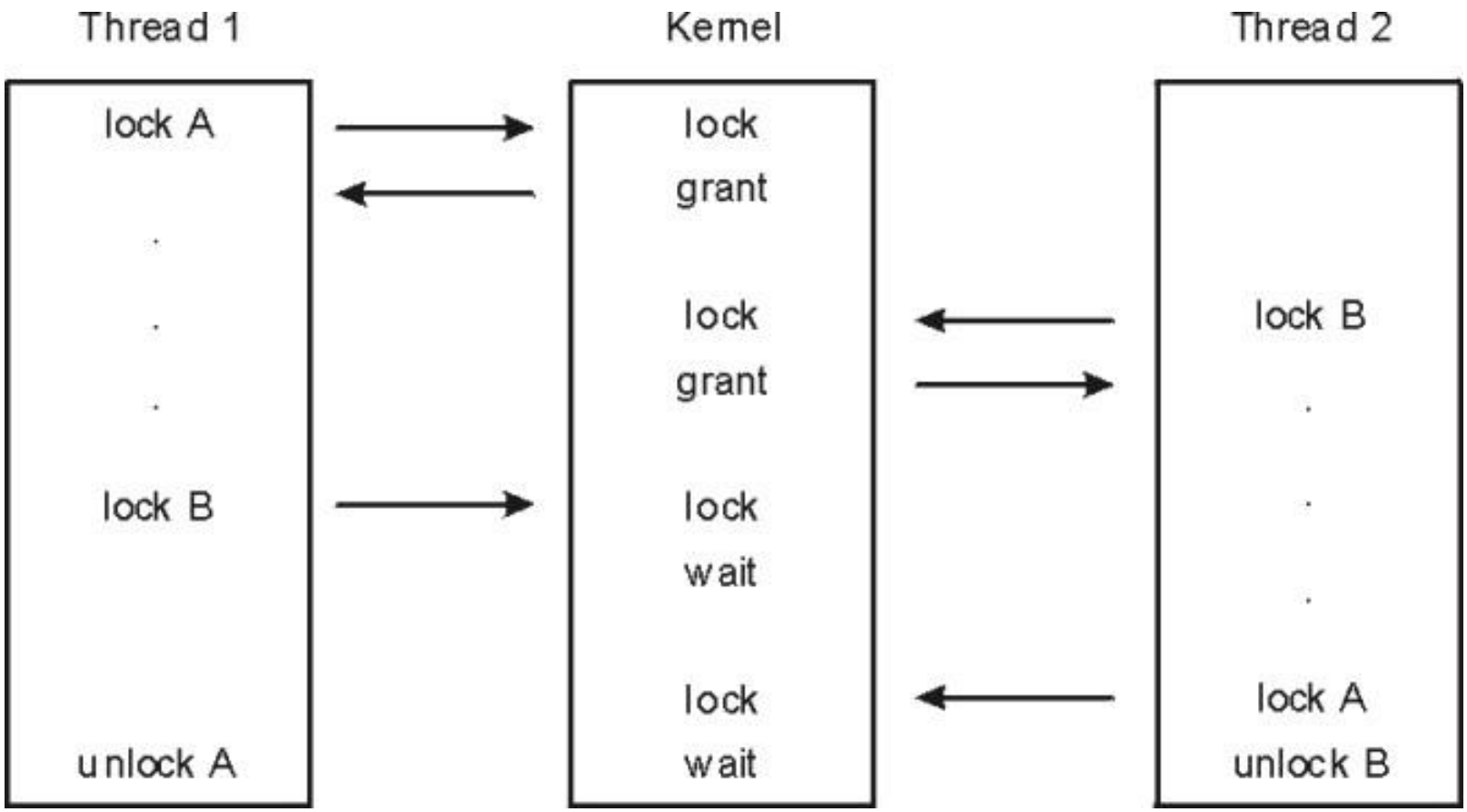
Deadlock (bloqueo mutuo) potencial



Deadlock (bloqueo mutuo)



Deadlock (bloqueo mutuo)

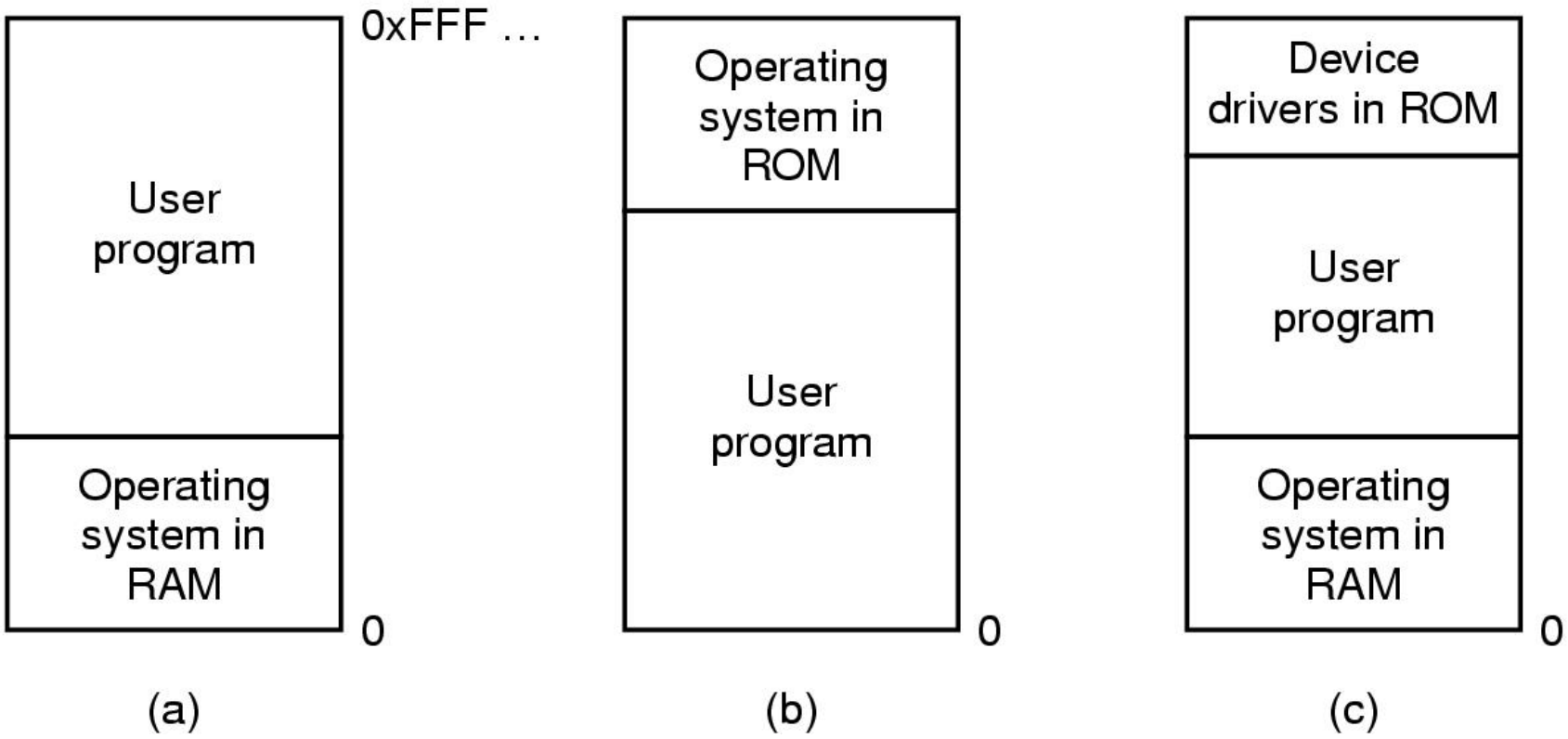


Estrategias para evitar deadlocks

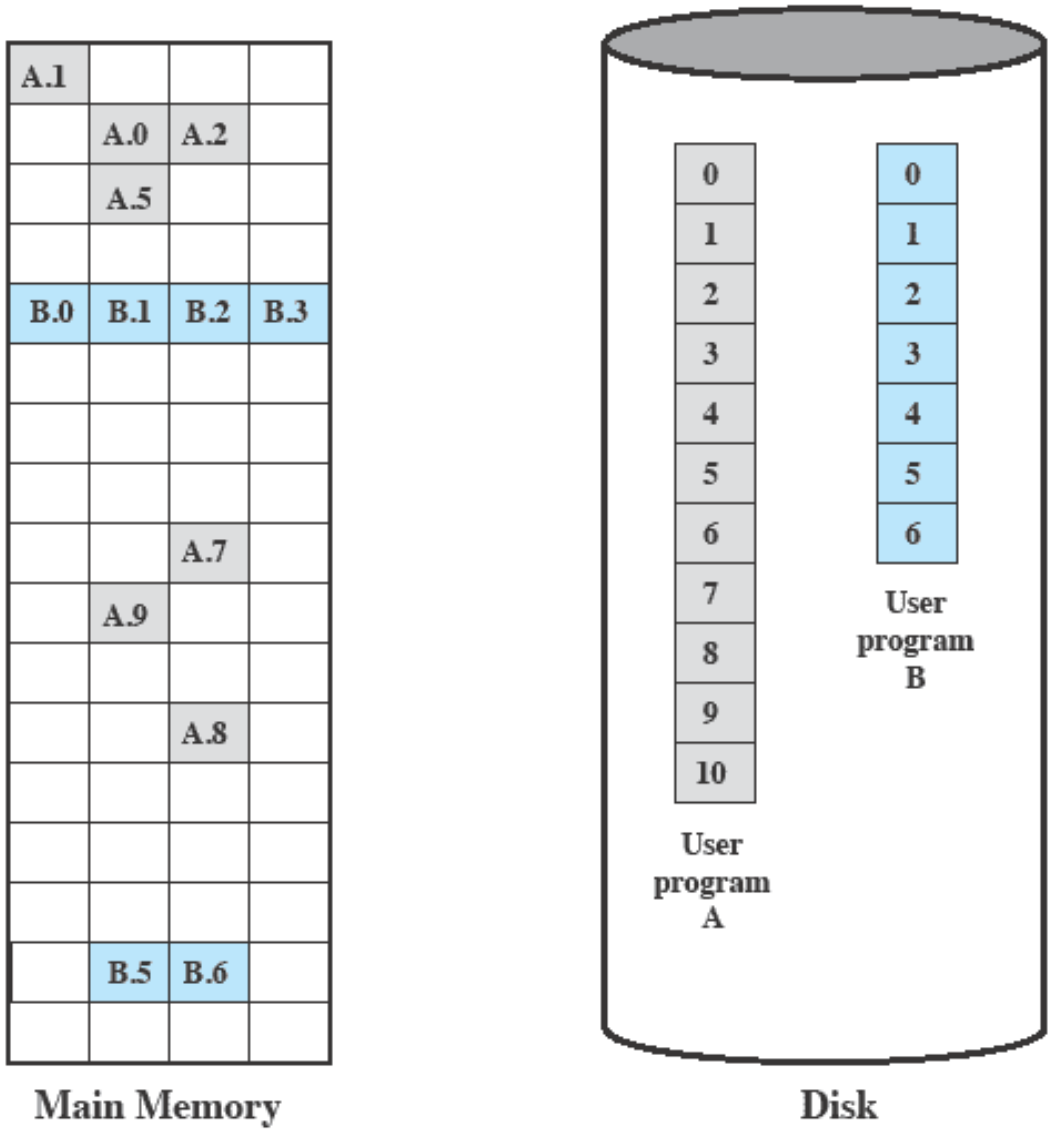
- Ignorar el problema (no muy recomendable)
- Detectar y Recuperación
- Prevenirlos y evitarlos dinámicamente
  - Semáforos, zonas críticas, métodos sincronizados.

Manejo de memoria

Modelos básicos: Monoprogramación

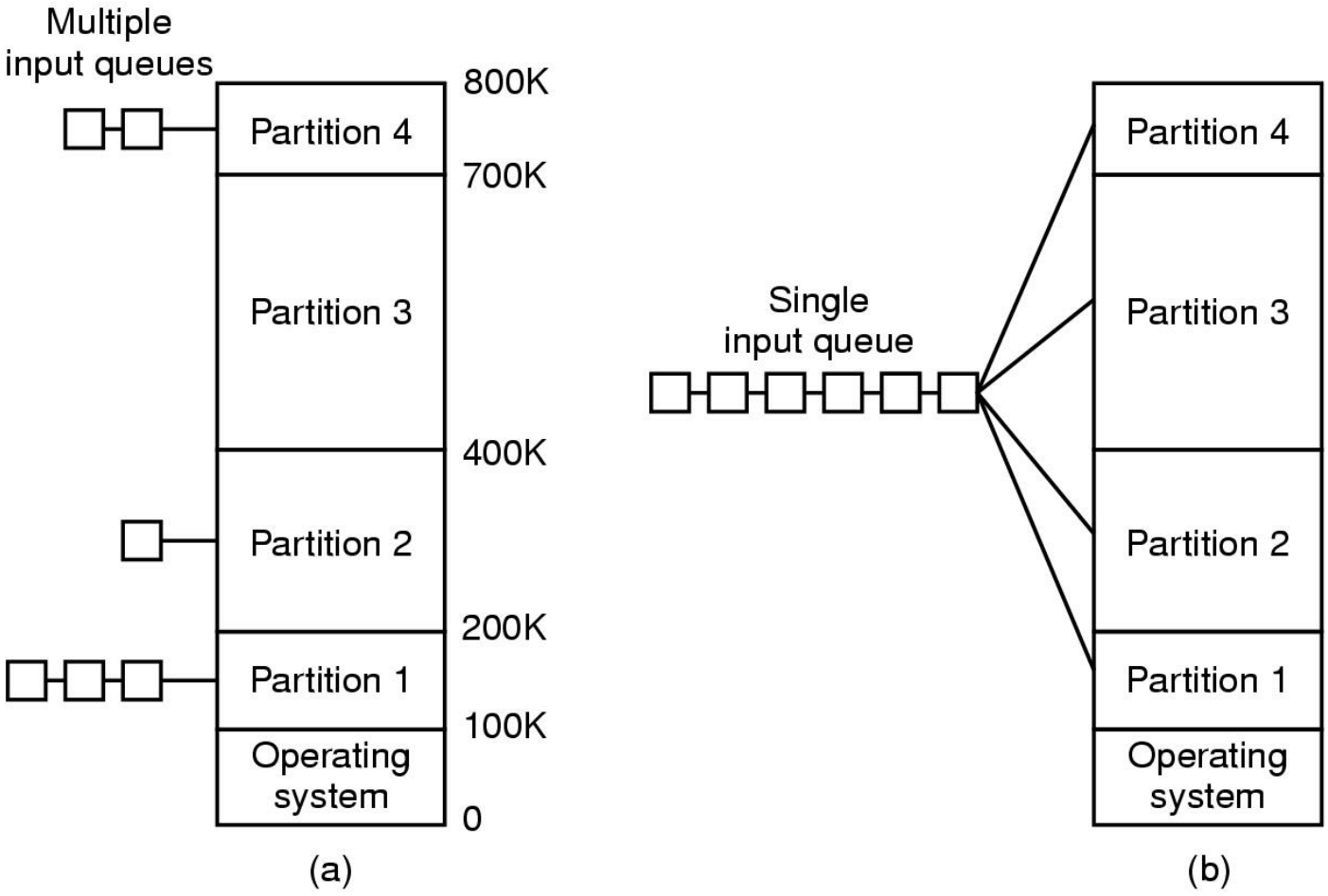


Manejo de memoria (Virtual)

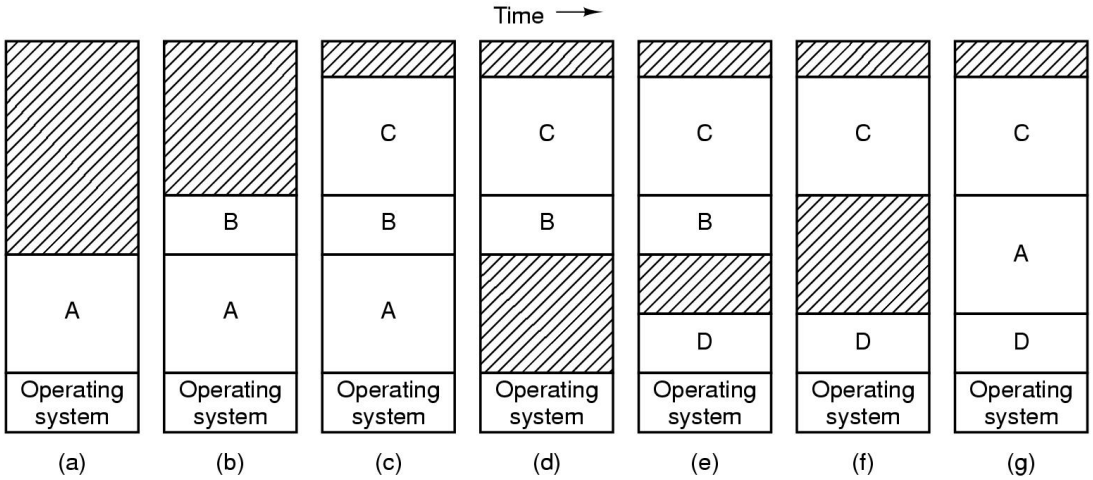


Manejo de memoria

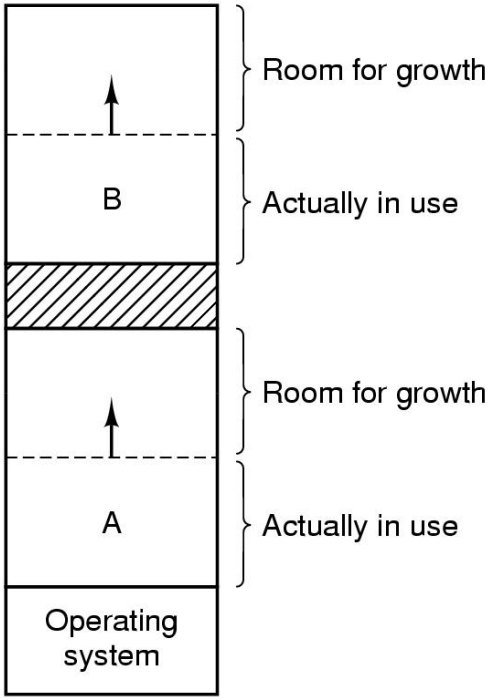
Modelos avanzados: Multiprogramación



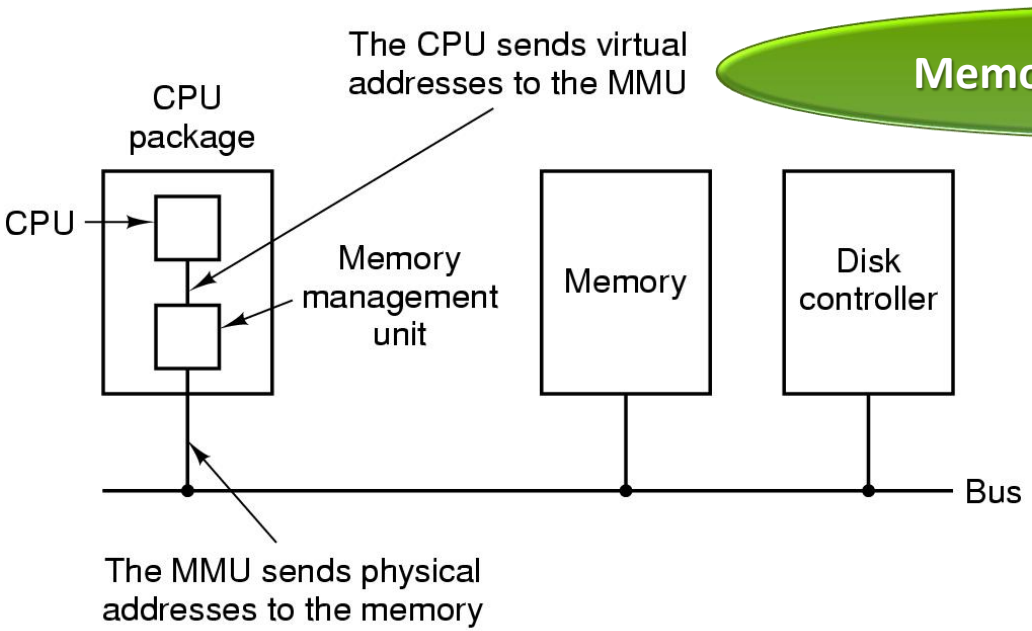
Swapping (Intercambio)



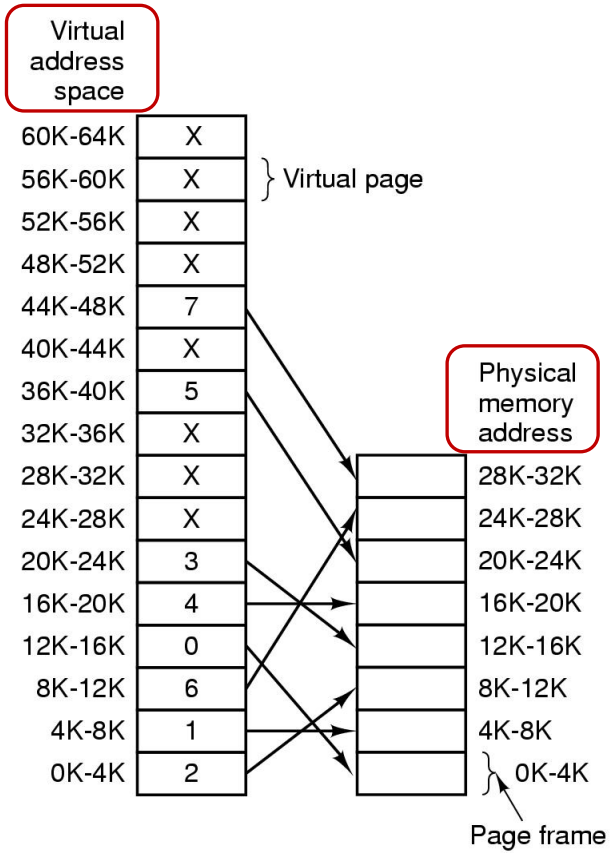
A, B, C, and D are processes



Memoria virtual - Paginación

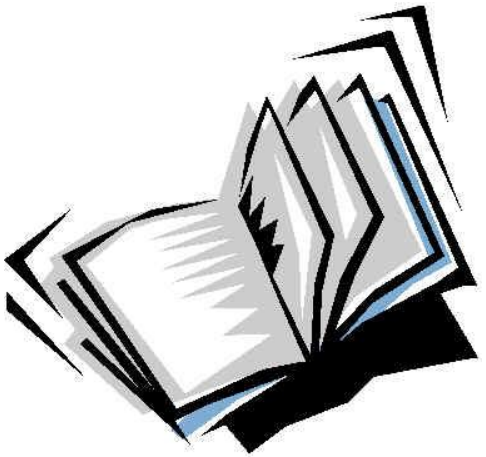


Memory Management Unit



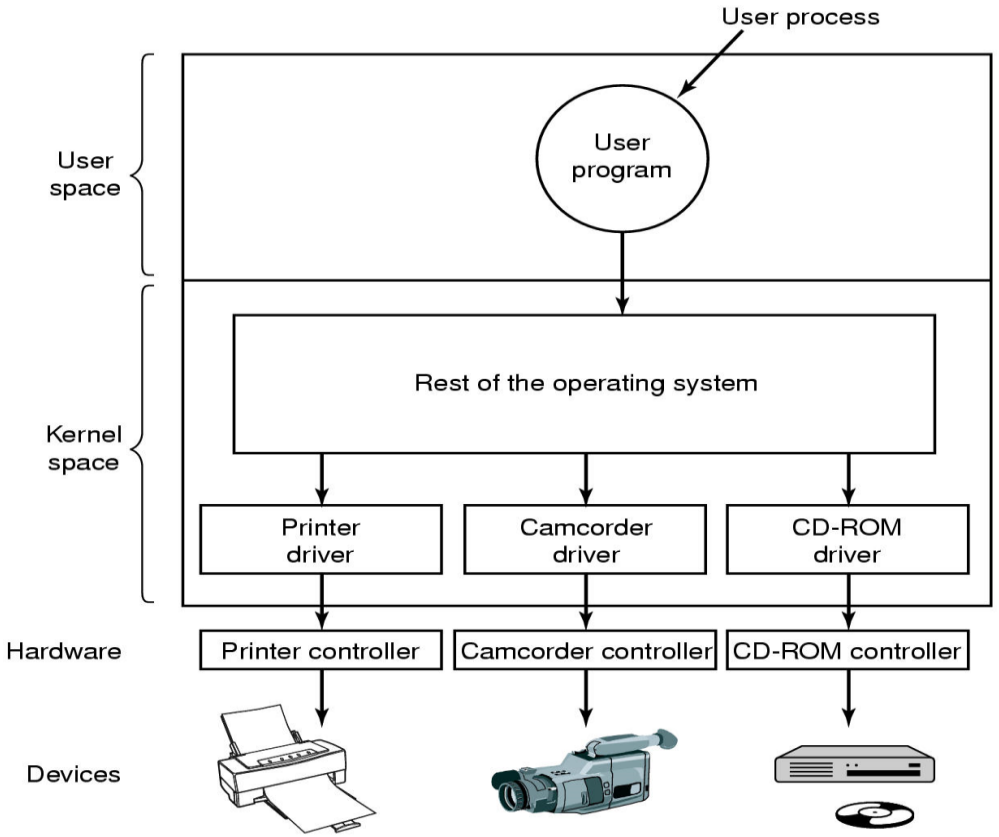
Algoritmos de reemplazo de páginas

- No usadas recientemente
- Menos usadas recientemente
- FIFO - Primera en entrar, primera en salir
- FIFO - Segunda oportunidad (con base en accesos previos)

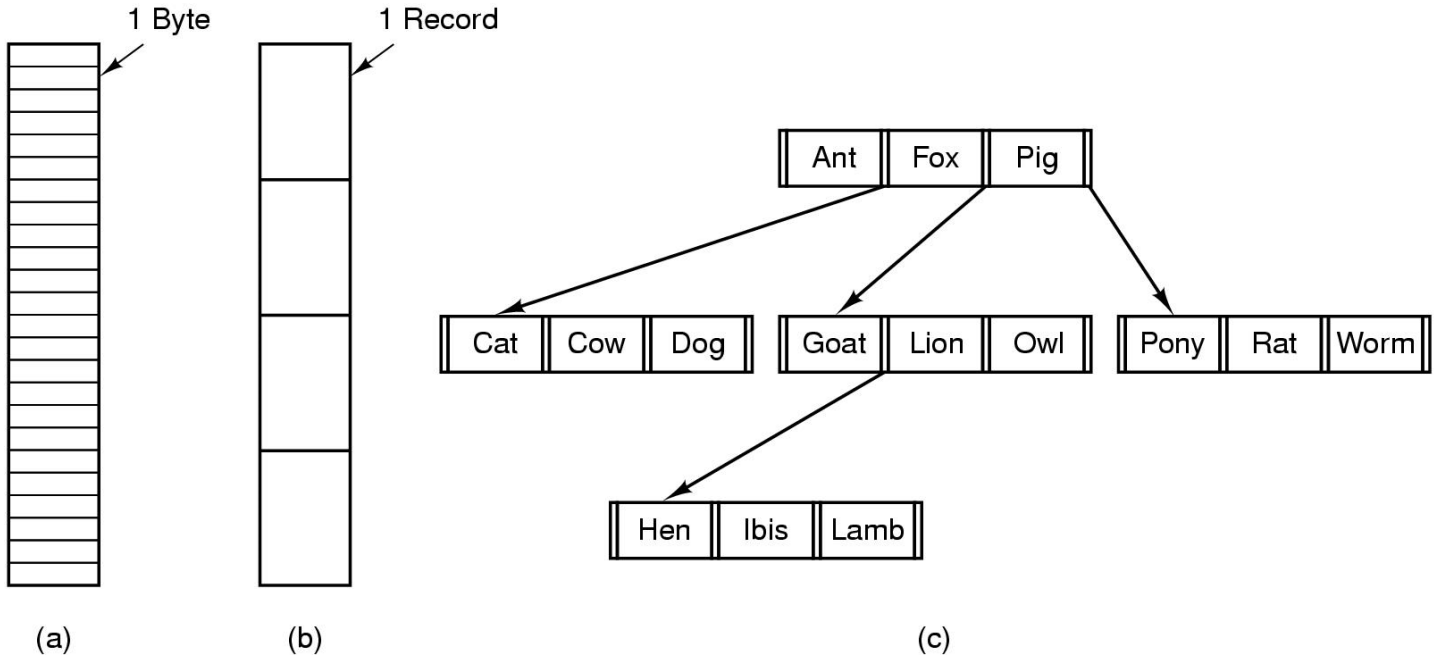


Entrada y Salida

Device	Data rate
Keyboard	10 bytes/sec
Mouse	100 bytes/sec
56K modem	7 KB/sec
Telephone channel	8 KB/sec
Dual ISDN lines	16 KB/sec
Laser printer	100 KB/sec
Scanner	400 KB/sec
Classic Ethernet	1.25 MB/sec
USB (Universal Serial Bus)	1.5 MB/sec
Digital camcorder	4 MB/sec
IDE disk	5 MB/sec
40x CD-ROM	6 MB/sec
Fast Ethernet	12.5 MB/sec
ISA bus	16.7 MB/sec
EIDE (ATA-2) disk	16.7 MB/sec
FireWire (IEEE 1394)	50 MB/sec
XGA Monitor	60 MB/sec
SONET OC-12 network	78 MB/sec
SCSI Ultra 2 disk	80 MB/sec
Gigabit Ethernet	125 MB/sec
Ultrium tape	320 MB/sec
PCI bus	528 MB/sec
Sun Gigaplane XB backplane	20 GB/sec



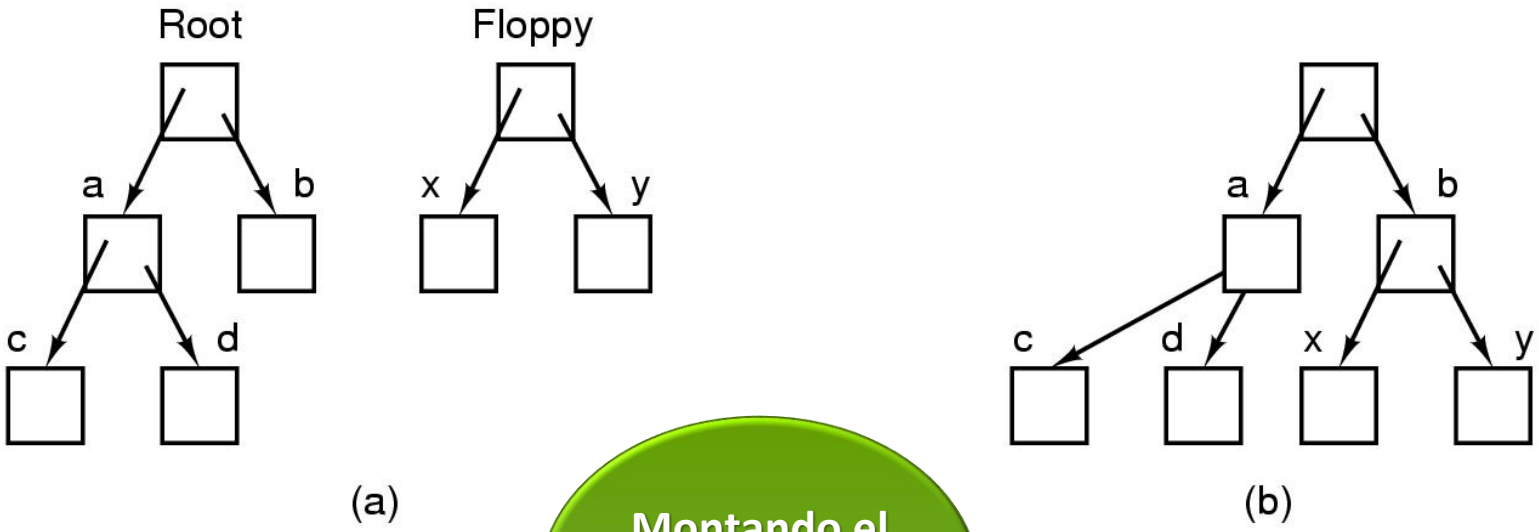
Sistema de archivos



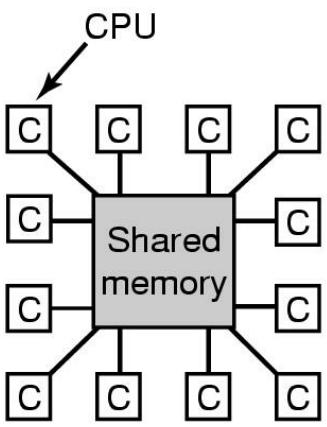
3 tipos de archivos: bytes, registros y árboles

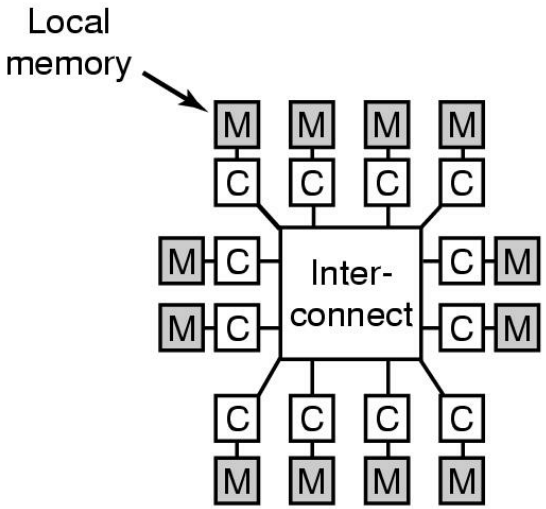
2 tipos de acceso: Secuencial y aleatorio

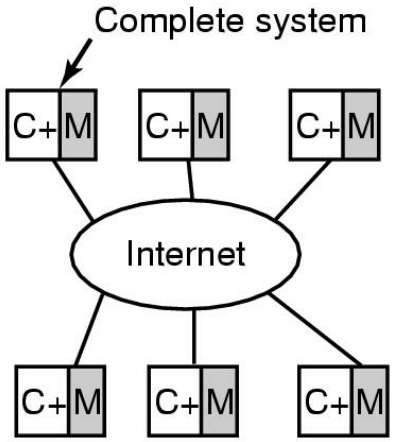
Sistema de archivos



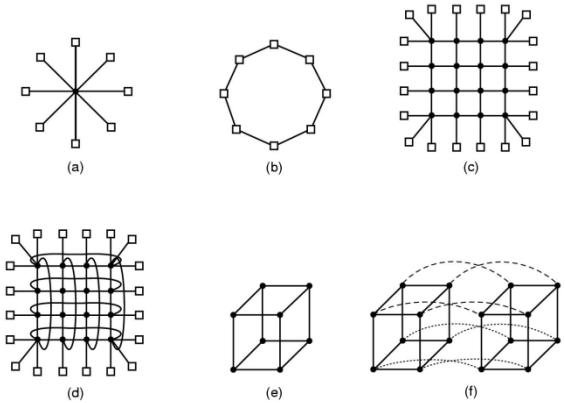
Sistemas de múltiples procesadores





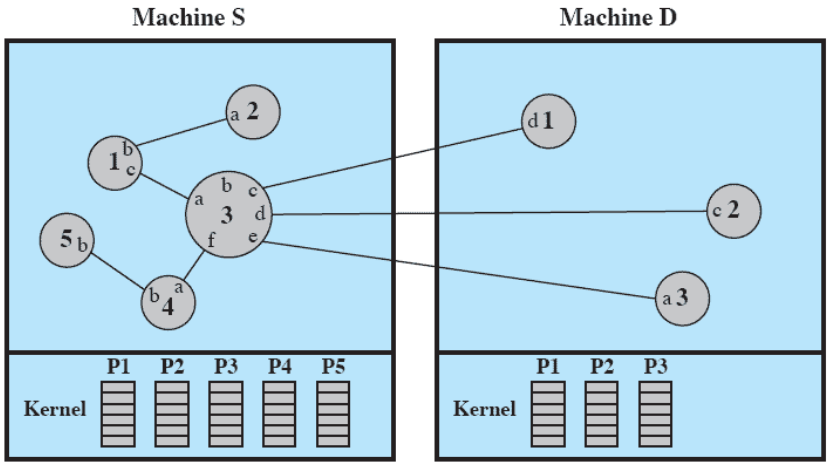


- Sincronización
- Balanceo de cargas
- Middlewares

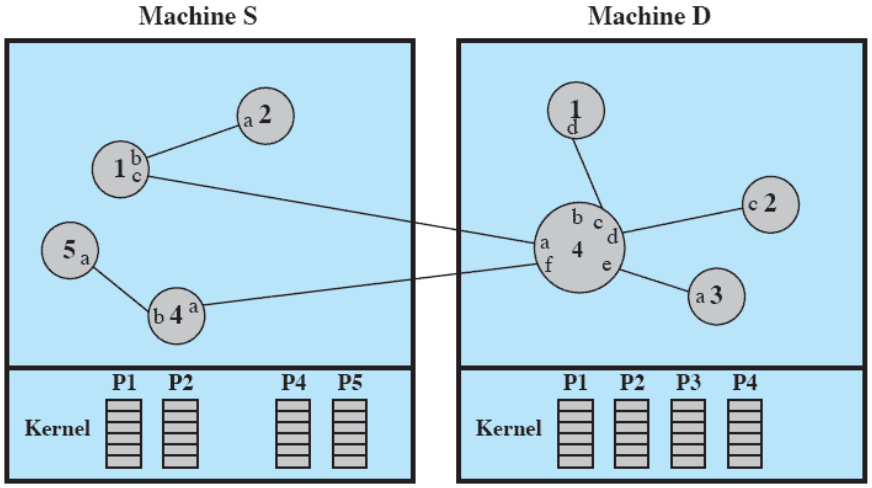


- Comunicación:
  - -Buses
  - - Mensajes
  - - RPC y RMI
  - - Memoria distribuida compartida

Migración de Procesos



(a) Before migration



(b) After migration