

Planificación de CPU

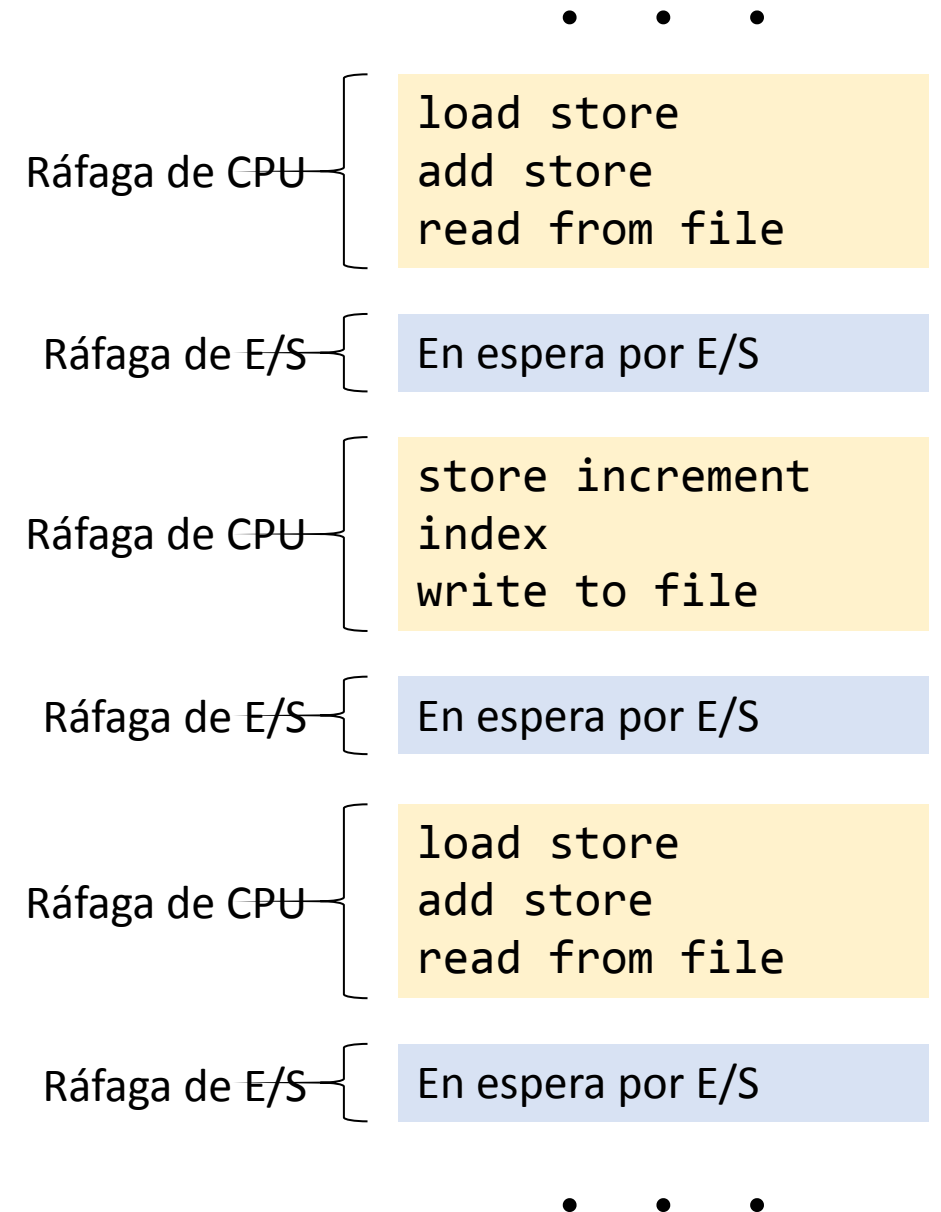
Adaptación de diferentes referencias (ver al final)

Conceptos

- Un solo procesador (o núcleo) significa un solo proceso a la vez en ejecución.
- Grado de multiprogramación
 - Número de procesos activos (**en memoria principal**) que mantiene un sistema
 - A más procesos activos mayor probabilidad de encontrar un proceso listo para ejecutar
 - Afecta de forma importante el rendimiento de un computador
- Objetivo de la multiprogramación
 - Tener **algún proceso** ejecutándose en todo momento para maximizar el uso de CPU

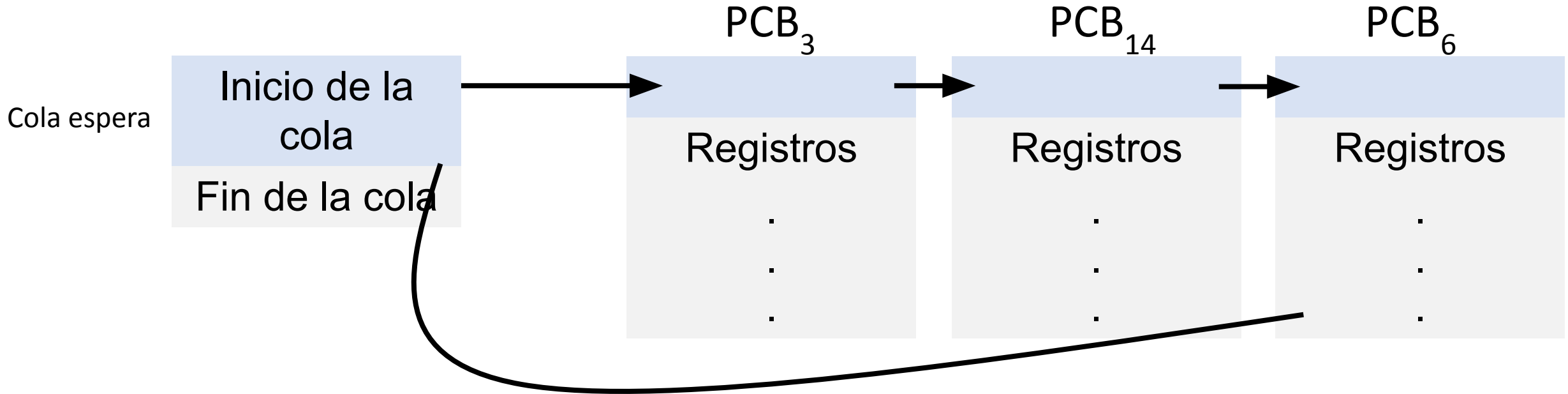
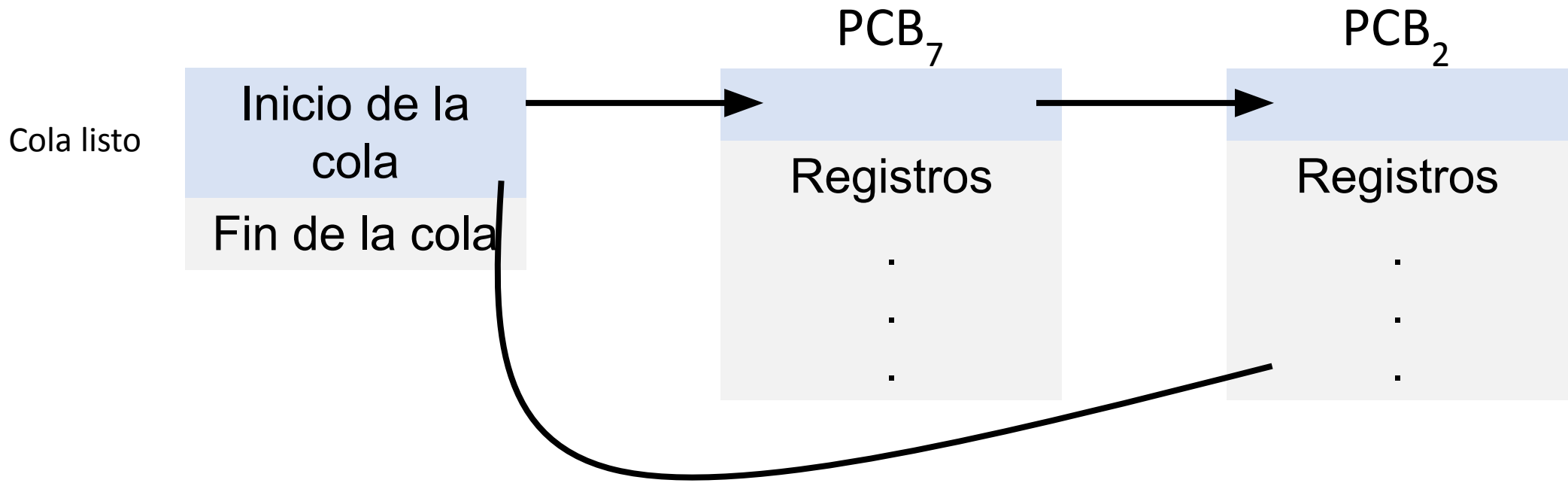
Ráfagas de CPU y de E/S

- La ejecución de un proceso consiste en alternar entre dos estados
 - En ejecución (**ráfaga de CPU**)
 - En espera por E/S (**ráfaga de E/S**)
- Patrón cíclico de ejecución de un proceso
- Objetivo de multiprogramación
 - Usar los tiempos de espera de E/S de un proceso para darle la CPU a otro proceso



Planificador de CPU

- Forma parte del núcleo del S.O
- Entra en ejecución cada vez que se activa el S.O
- Selecciona el siguiente proceso a ejecutar
 - Lo selecciona de la cola de **listo**.
 - La cola de listo puede atenderse con diferentes disciplinas.
 - **No** necesariamente es FIFO.



Planificación apropiativa y no apropiativa

- **Circunstancias** en las que entra el planificador a tomar decisiones

1. Proceso pasa de **ejecución** a **espera** (E/S o llamada a wait)
2. Proceso pasa de **ejecución** a **listo** (ocurre una interrupción)
3. Proceso pasa de **espera** a **listo** (se completa una operación de E/S)
4. Proceso termina



Planificación apropiativa y no apropiativa

- **Circunstancias 1 y 4**

- No hay decisiones en términos de planificación
- Siempre que haya un proceso en estado de listo, él es el que sigue en CPU
- **Planificación NO apropiativa**
- El proceso permanece en CPU hasta que termine o hasta que pase a espera

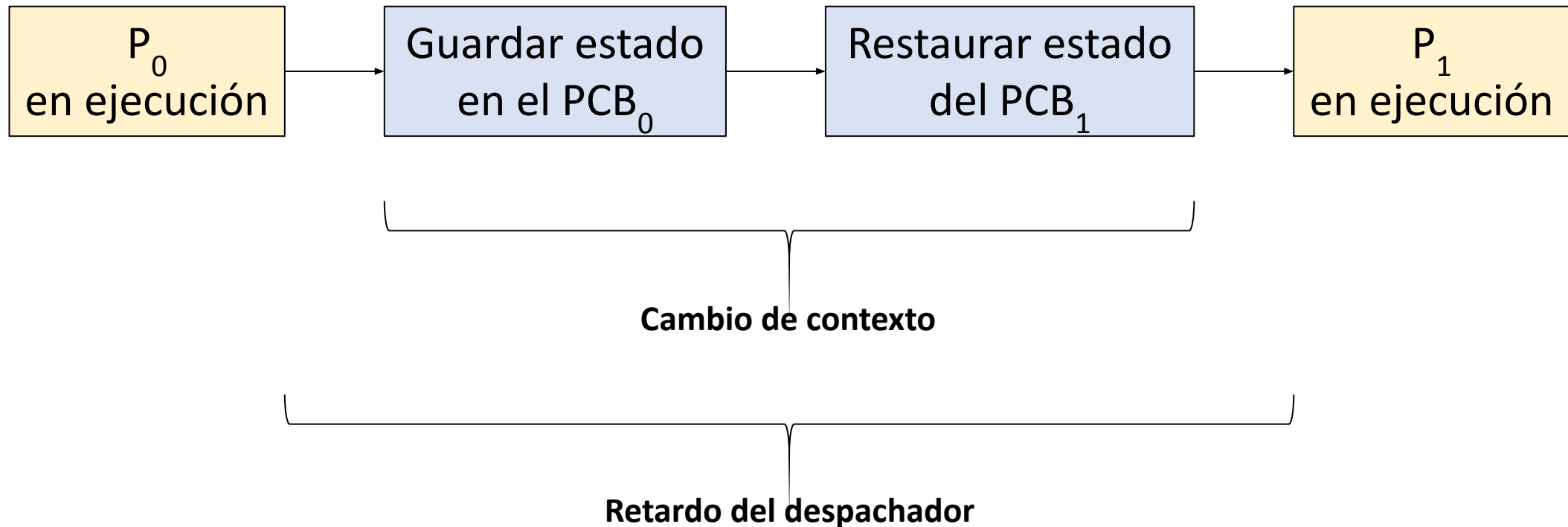
- **Circunstancias 2 y 3**

- Si hay decisiones en términos de planificación
- ¿A quién se le asigna la CPU? ¿Al proceso interrumpido? ¿A otro? ¿Se interrumpe el que ya está en ejecución?
- **Planificación apropiativa**
- Se dan condiciones de carrera: un proceso actualizando datos (compartidos) y entra otro

Despachador

- Forma parte del núcleo del S.O
- Responsable de entregarle a la CPU el proceso seleccionado por el planificador
 - Realiza el cambio de contexto
 - Cambia el modo de ejecución: pasa de modo kernel (**ring 0**) a modo usuario (**ring 3**)
 - Apunta el registro IP a la siguiente instrucción del proceso que será ejecutado
- Estas operaciones introducen un retardo
 - Retardo del despachador
 - Se desea que sea lo más rápido posible

Despachador



Criterios (objetivos) de planificación

- **Utilización CPU**

- Mantener la CPU lo más ocupada posible: ejecutando algún proceso.

- **Throughput**

- Número de procesos que se completan por unidad de tiempo.

- **Tiempo de ida y vuelta**

- Suma de tiempos: de espera en cola de listo + tiempo en CPU + tiempo en E/S

- **Tiempo de espera**

- Suma de tiempos en los que proceso pasa esperando en la cola de listo.

- **Tiempo de respuesta**

- Tiempo que le toma al proceso comenzar a responder después de recibir una solicitud

Criterios (objetivos) de planificación

- **Deseable maximizar**

- Uso de CPU
- Throughput

- **Deseable minimizar**

- Tiempo de ida y vuelta
- Tiempo de espera
- Tiempo de respuesta

POSIX

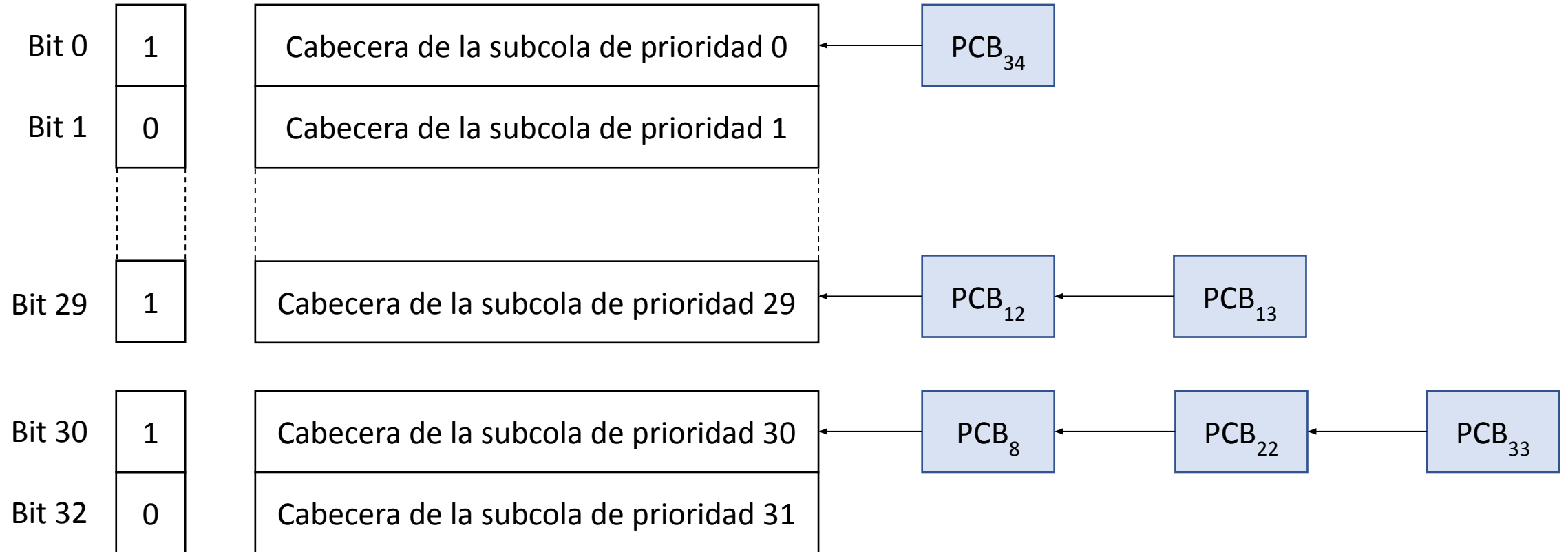
- POSIX es una estándar de IEEE
 - Recomendación de una interfaz estándar del S.O
 - También como Single Unix Specification SUS (hoy SUSv4 POSIX.1-2008)
- Incluye
 - Interfaz estándar del S.O y el entorno
 - Interpretes de comandos
 - Programas y utilidades comunes
- Propósito
 - Apoyar la portabilidad de aplicaciones a nivel de código fuente
- P. Ej.: POSIX.1
 - Biblioteca estándar de C, creación y control de procesos, etc.

Planificación en POSIX

- Cada proceso (o hilo) lleva asociada
 - Una política de planificación
 - Una prioridad
- Cada política de planificación lleva asociada un rango de prioridades
 - Al menos 32 niveles de prioridad según estándar POSIX
 - El planificador selecciona siempre el proceso con la prioridad más alta
- Linux/Unix
 - **40** niveles de prioridad: desde **-20** (la más alta) a **19** (la más baja)
 - Procesos iniciados por usuarios se les asigna prioridad 0
 - El comando **nice** sirve para modificar la prioridad de un proceso en ejecución

Colas de planificación

Palabra resumen



Políticas de planificación en POSIX

- **FIFO**

1. Procesos se agregan al final de la cola de su prioridad asociada
2. Proceso se expulsa de CPU cuando ejecute llamada bloqueante
3. Proceso se expulsa de CPU cuando aparezca un proceso con mayor prioridad

- **Reglas del planificador para FIFO**

1. Proceso expulsado por causa No. 3: proceso expulsado pasa a ser el primero de la cola de su prioridad
2. Proceso pasa de bloqueado a listo: proceso se agrega al final de la cola de su prioridad
3. Cambio de prioridad o política: se realiza replanificación, si resulta expulsado, se agrega al final de la cola

Políticas de planificación en POSIX

- Cíclica

- Se asigna rodaja de tiempo (*quantum*) a procesos en colas de prioridad
- Proceso que acaba su *quantum* se agrega al final de la cola de su prioridad
- Proceso expulsado por otro de mayor prioridad: expulsado se agrega al principio de la cola pero sin restaurar su *quantum*.

Planificación en Windows

- La unidad fundamental de planificación en Windows es el hilo
- Se usan 32 niveles de prioridades en planificación cíclica
 1. Dieciséis niveles para procesos en tiempo real: del 16 al 31
 2. Quince niveles variables: del 1 al 15
 3. Un nivel para el sistema: 0
- Procesos en mismo nivel reciben el mismo *quantum*.
- Procesos en nivel dos:
 - Inician con una prioridad determinada y ésta va cambiando pero sin llegar a nivel 16.
 - Prioridad disminuye si acaba su *quantum*
 - Prioridad aumenta si proceso se bloquea por E/S

Planificación en Windows

- Lectura complementaria
 - Processes, Threads, and Jobs in the Windows Operating System
 - <https://www.microsoftpressstore.com/articles/article.aspx?p=2233328&seqNum=7>

Referencias

- Carretero Pérez, J., García Carballeira, F., de Miguel Anasagasti, P., & Pérez Costoya, F. (2001). Planificación. In *Sistemas operativos. Una Visión Aplicada* (pp. 102–109). McGraw Hill.
- Silberschatz, A., Baer Galvin, P., & Gagne, G. (2018). CPU Scheduling. In *Operating Systems Concepts* (10th ed., pp. 199–205). John Wiley & Sons, Inc.