

Planificación

Planificación de procesos

Eloy Anguiano Rey
eloy.anguiano@uam.es

Ana González
ana.marcos@uam.es

Escuela Politécnica Superior
Universidad Autónoma de Madrid



Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Parte I

Planificación de un sólo procesador



Introducción

Objetivos de la planificación

Planificación de
procesos

Introducción

Objetivos de la
planificación

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Entre los objetivos básicos de la multiprogramación destacan:

- Mejora del tiempo de respuesta.
- Aumento de la productividad.
- Aumento de la eficiencia del procesador.

Tipos de planificación

Planificación de procesos

Introducción

Tipos de planificación

Planificación en el diagrama de estados

Niveles de planificación

Criterios de planificación

Planificación a largo plazo

Planificación a medio plazo

Planificación a corto plazo

Criterios de planificación a corto plazo

Prioridades

Políticas de planificación

Algoritmos de planificación

Otras políticas

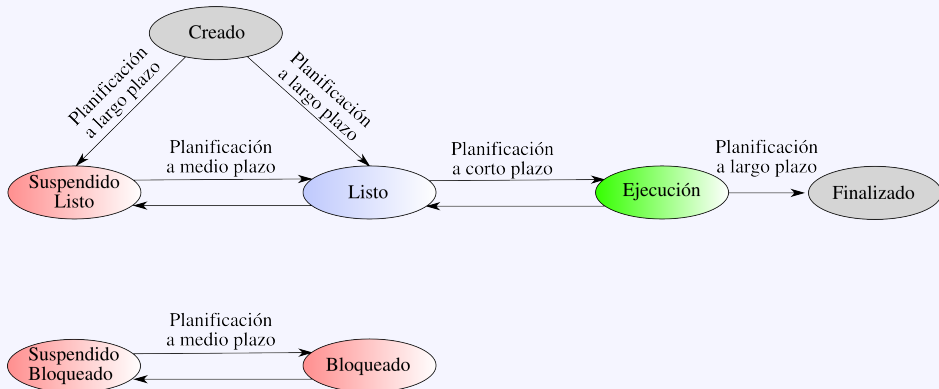
- **Planificación a largo plazo:** decisión de añadir procesos al conjunto de procesos a ejecutar.
- **Planificación a medio plazo:** decisión de añadir procesos al conjunto de procesos que se encuentran parcial o completamente en memoria.
- **Planificación a corto plazo:** decisión sobre qué proceso disponible será ejecutado en el procesador.
- **Planificación de E/S:** decisión sobre qué solicitud de E/S pendiente será tratada por un dispositivo de E/S disponible.

Tipos de planificación

Planificación en el diagrama de estados

Planificación de
procesos

Estados



Introducción

Tipos de
planificación

Planificación en el
diagrama de estados

Niveles de
planificación

Criterios de
planificación

Planificación a largo
plazo

Planificación a medio
plazo

Planificación a corto
plazo

Criterios de
planificación a corto
plazo

Prioridades

Políticas de
planificación

Algoritmos de
planificación

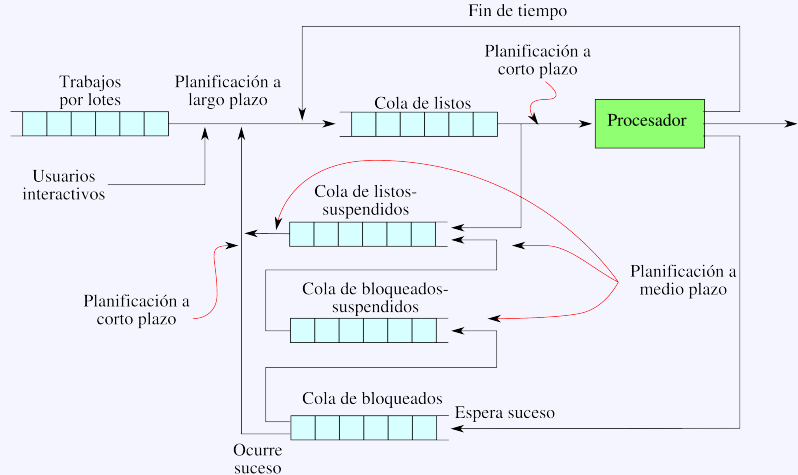
Otras políticas

Tipos de planificación

Planificación en el diagrama de estados

Planificación de
procesos

Colas



Introducción

Tipos de
planificación

Planificación en el
diagrama de estados

Niveles de
planificación

Criterios de
planificación

Planificación a largo
plazo

Planificación a medio
plazo

Planificación a corto
plazo

Criterios de
planificación a corto
plazo

Prioridades

Políticas de
planificación

Algoritmos de
planificación

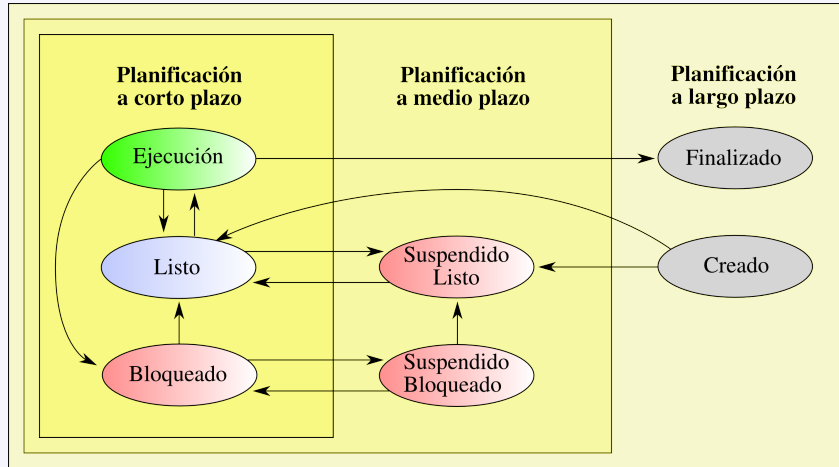
Otras políticas

Tipos de planificación

Niveles de planificación

Planificación de
procesos

Colas



Introducción

Tipos de
planificación

Planificación en el
diagrama de estados

Niveles de
planificación

Criterios de
planificación

Planificación a largo
plazo

Planificación a medio
plazo

Planificación a corto
plazo

Criterios de
planificación a corto
plazo

Prioridades

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

Tipos de planificación

Criterios de planificación

Los criterios de planificación se deciden en función de las siguientes cantidades (relacionadas con el ejemplo):

- **Uso de CPU (%)**: $U_{CPU} = 100 \frac{T_t - (t_2 - t_1)}{T_t}$
- **Rendimiento (pr/ut)**: $R = 2 / T_t$
- **Tiempo de retorno medio**: $T_{rm} = \frac{t_{11} + (T_t - t_3)}{2}$
- **Tiempo de espera medio**:

$$\frac{(t_6 - t_5) + (t_9 - t_8) + (t_4 - t_3) + (t_7 - t_6) + (t_{11} - t_{10})}{2}$$



- Ejecución
- Listo
- Espera E/S
- Sin cargar-Terminado

Tipos de planificación

Planificación a largo plazo

Planificación de
procesos

Introducción

Tipos de
planificación

Planificación en el
diagrama de estados

Niveles de
planificación

Criterios de
planificación

**Planificación a largo
plazo**

Planificación a medio
plazo

Planificación a corto
plazo

Criterios de
planificación a corto
plazo

Prioridades

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

- Controla el grado de multiprogramación
 - Limitar el número para dar buen servicio
 - **Nuevo**: cada vez que termina un proceso o si el porcentaje de utilización del procesador es bajo
- Determina cuáles son los programas admitidos en el sistema
 - Algoritmos de planificación
 - Simples (ej., FIFO-FCFS)
 - Por rendimiento del sistema: prioridades, carga procesador, carga E/S, recurso E/S a solicitar, ...
- Cuantos más procesos se crean, menor es el porcentaje de tiempo en el que cada proceso se puede ejecutar

Tipos de planificación

Planificación a medio plazo

Planificación de
procesos

Introducción

Tipos de planificación

Planificación en el
diagrama de estados

Niveles de
planificación

Criterios de
planificación

Planificación a largo
plazo

**Planificación a medio
plazo**

Planificación a corto
plazo

Criterios de
planificación a corto
plazo

Prioridades

Políticas de planificación

Algoritmos de planificación

Otras políticas

- Forma parte de la función de intercambio
 - Gestión de memoria, Memoria Virtual, Estados Suspendidos
- Se basa en la necesidad de controlar el grado de multiprogramación

Tipos de planificación

Planificación a corto plazo

Planificación de
procesos

Introducción

Tipos de planificación

Planificación en el
diagrama de estados

Niveles de
planificación

Criterios de
planificación

Planificación a largo
plazo

Planificación a medio
plazo

**Planificación a corto
plazo**

Criterios de
planificación a corto
plazo

Prioridades

Políticas de planificación

Algoritmos de planificación

Otras políticas

También conocido como distribuidor o “dispatcher”:

- Es el de ejecución más frecuente.
- Se ejecuta cuando ocurre un suceso de entre los siguientes:
 - Interrupciones del reloj.
 - Interrupciones de E/S.
 - Llamadas al sistema operativo.
 - Señales.

Tipos de planificación

Criterios de planificación a corto plazo

Planificación de
procesos

Introducción

Tipos de
planificación

Planificación en el
diagrama de estados

Niveles de
planificación

Criterios de
planificación

Planificación a largo
plazo

Planificación a medio
plazo

Planificación a corto
plazo

Criterios de
planificación a corto
plazo

Prioridades

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

Orientados al usuario

Cuantitativos

- Tiempo de retorno
 - Desde el lanzamiento hasta la finalización de un proceso.
 - Apropiado para trabajos por lotes
- Tiempo de respuesta
 - Desde que se emite solicitud hasta que la respuesta aparece en la salida.
 - Apropiaada para procesos interactivos
- Plazos
 - Si hay plazos, maximizar porcentaje de plazos cumplidos
 - Caminos críticos: a seguir si se quieren cumplir los requisitos

Cualitativos

- Previsibilidad
 - Tiempo y coste independiente de la carga del sistema

Tipos de planificación

Criterios de planificación a corto plazo

Planificación de
procesos

Introducción

Tipos de
planificación

Planificación en el
diagrama de estados

Niveles de
planificación

Criterios de
planificación

Planificación a largo
plazo

Planificación a medio
plazo

Planificación a corto
plazo

Criterios de
planificación a corto
plazo

Prioridades

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

Orientados al sistema

Cuantitativos

- Productividad
 - Maximizar n° procesos / unidad de tiempo
- Utilización del procesador
 - Importante en sistemas compartidos caros
 - Menos importante en monousuario y en tiempo real

Cualitativos

- Equidad
- No inanición
- Prioridades: si hay, favorecer a procesos con mayor
- Equilibrio de ocupación de recursos
 - Mantener ocupados los recursos
 - Favorecer procesos que no usen recursos sobrecargados

Tipos de planificación

Prioridades

Planificación de procesos

Introducción

Tipos de planificación

Planificación en el diagrama de estados

Niveles de planificación

Criterios de planificación

Planificación a largo plazo

Planificación a medio plazo

Planificación a corto plazo

Criterios de planificación a corto plazo

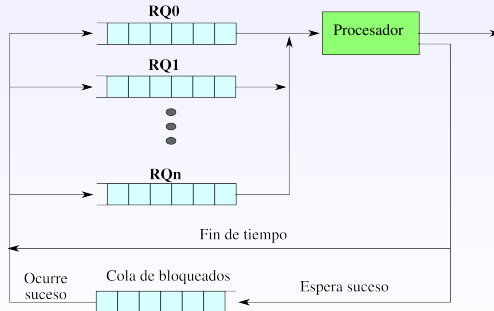
Prioridades

Políticas de planificación

Algoritmos de planificación

Otras políticas

- El planificador seleccionará siempre a un proceso de mayor prioridad antes que a los de menor prioridad.
- Tiene múltiples colas de Listos para representar cada nivel de prioridad.
- Los procesos de prioridad más baja pueden sufrir inanición.
- Permite que un proceso cambie su prioridad en función de su edad o su historial de ejecución.



Políticas de planificación

Función de selección

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Función de selección
Modo de selección

Algoritmos de
planificación

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Cómo se selecciona el siguiente proceso a ejecutar

Es necesario tener en cuenta múltiples características. Por ejemplo:

- Prioridades
- Necesidades de recursos
- Características de ejecución:
 - Tiempo en el sistema
 - Tiempo ejecutado
 - Tiempo total estimado

Políticas de planificación

Modo de selección

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Función de selección
Modo de selección

Algoritmos de
planificación

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Momento en que se aplica la función de selección

- **No preferente, no expulsiva, (apropiativa):**
 - Una vez que el proceso pasa al estado de Ejecución, continúa ejecutando hasta que termina, se bloquea en espera de una E/S o solicita el servicio del SO.
- **Preferente, expulsiva, (no apropiativa):**
 - El proceso que se está ejecutando actualmente puede ser interrumpido y pasado al estado de Listos por el sistema operativo.
 - Nuevo proceso
 - Proceso de mayor prioridad sale de bloqueado
 - Interrupción de reloj
 - Permiten dar un mejor servicio ya que evitan que un proceso pueda monopolizar el procesador durante mucho tiempo.
 - Mayor coste: más cambios de contexto

Algoritmos de planificación

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Existen múltiples algoritmos de planificación. En esta sección vamos a ver:

- FCFS (First-come, First-served)
- Turno rotatorio (Round-Robin)
- SPN (Shortest Process Next)
- SRT (Shortest Remaining Time)
- HRRN (Highest Response Ratio Next)
- Realimentación
- Reparto equitativo
- Planificación garantizada

Algoritmos de planificación

FCFS

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

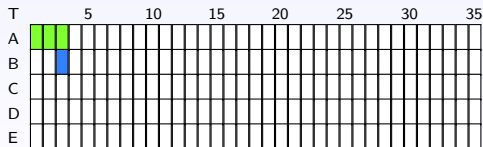
Evaluación de
algoritmos

Criterio de llegada inicial

Servicio por orden de llegada (First Come First Served) FCFS. Cuando un proceso solicita uso de CPU (pasa a estar listo para ejecución) su BCP se pone el último en una cola (FIFO) de los procesos en espera de tiempo de CPU. Es un algoritmo **apropiativo**.

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Continúa ejecutándose
Llega nuevo



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Algoritmos de planificación

FCFS

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

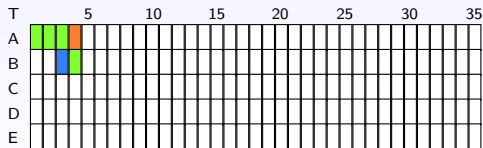
Evaluación de
algoritmos

Criterio de llegada inicial

Servicio por orden de llegada (First Come First Served) FCFS. Cuando un proceso solicita uso de CPU (pasa a estar listo para ejecución) su PCB se pone el último en una cola (FIFO) de los procesos en espera de tiempo de CPU. Es un algoritmo **apropiativo**.

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Se bloquea
Se ejecuta



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Algoritmos de planificación

FCFS

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

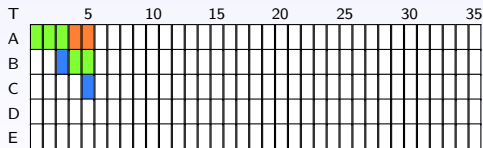
Evaluación de
algoritmos

Criterio de llegada inicial

Servicio por orden de llegada (First Come First Served) FCFS. Cuando un proceso solicita uso de CPU (pasa a estar listo para ejecución) su PCB se pone el último en una cola (FIFO) de los procesos en espera de tiempo de CPU. Es un algoritmo **apropiativo**.

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Llega nuevo



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Algoritmos de planificación

FCFS

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

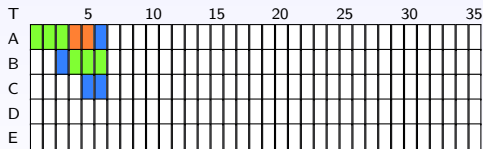
Evaluación de
algoritmos

Criterio de llegada inicial

Servicio por orden de llegada (First Come First Served) FCFS. Cuando un proceso solicita uso de CPU (pasa a estar listo para ejecución) su PCB se pone el último en una cola (FIFO) de los procesos en espera de tiempo de CPU. Es un algoritmo **apropiativo**.

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Se desbloquea



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Algoritmos de planificación

FCFS

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

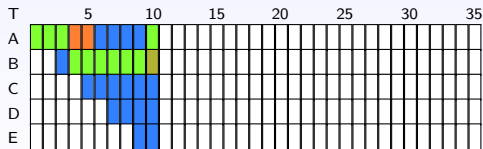
Planificación en
UNIX

Evaluación de
algoritmos

Criterio de llegada inicial

Servicio por orden de llegada (First Come First Served) FCFS. Cuando un proceso solicita uso de CPU (pasa a estar listo para ejecución) su PCB se pone el último en una cola (FIFO) de los procesos en espera de tiempo de CPU. Es un algoritmo **apropiativo**.

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Algoritmos de planificación

FCFS

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

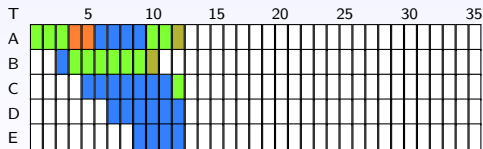
Evaluación de
algoritmos

Criterio de llegada inicial

Servicio por orden de llegada (First Come First Served) FCFS. Cuando un proceso solicita uso de CPU (pasa a estar listo para ejecución) su PCB se pone el último en una cola (FIFO) de los procesos en espera de tiempo de CPU. Es un algoritmo **apropiativo**.

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Termina
Se ejecuta



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Algoritmos de planificación

FCFS

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

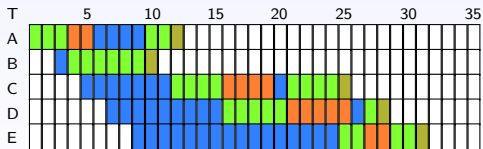
Planificación en
UNIX

Evaluación de
algoritmos

Criterio de llegada inicial

Servicio por orden de llegada (First Come First Served) FCFS. Cuando un proceso solicita uso de CPU (pasa a estar listo para ejecución) su PCB se pone el último en una cola (FIFO) de los procesos en espera de tiempo de CPU. Es un algoritmo **apropiativo**.

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

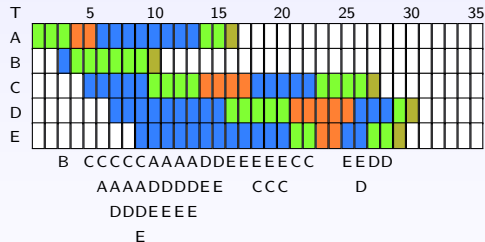
Algoritmos de planificación

FCFS

Planificación de
procesos

Criterio de llegada a la cola de listos

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

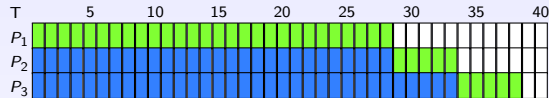
Evaluación de
algoritmos

Algoritmos de planificación

FCFS

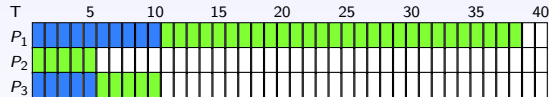
Planificación de
procesos

Es teóricamente justo, pero poco eficiente en tiempo de espera medio.



$$\text{Tiempo de espera medio} = \frac{28+33+38}{3} = 33$$

Un resultado mejor sería de la forma:



$$\text{Tiempo de espera medio} = \frac{5+10+38}{3} = 17,66$$

█ Ejecución █ Listo █ Espera E/S █ Sin cargar █ Terminado

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Algoritmos de planificación

FCFS

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

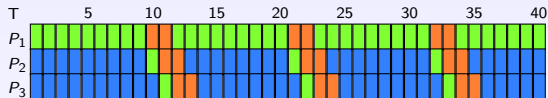
HRRN

Otras políticas

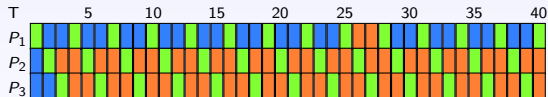
Planificación en
UNIX

Evaluación de
algoritmos

Efecto convoy, predominio de los procesos que usan CPU frente a los que usan E/S.



Un reparto más equitativo sería de la forma:



Se penaliza a los procesos más cortos. ■ Ejecución ■ Listo ■ Espera E/S ■ Sin cargar ■ Terminado

Algoritmos de planificación

Round-Robin, turno rotatorio

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

- Periódicamente, se genera una interrupción de reloj.
- Cuando se genera la interrupción, el proceso que está en ejecución se sitúa en la cola de Listos y se selecciona el siguiente trabajo (**no apropiativo**)
- Se conoce también como fracciones de tiempo.
- Está diseñado específicamente para sistemas de tiempo compartido. Se asigna un cuanto de tiempo (10-100 ms.) de igual duración a todos los procesos listos para ser ejecutados. Entre ellos, la selección se realiza mediante una cola FIFO.
- Parámetro crítico: **tamaño del cuanto**. La efectividad depende del tamaño del el cuanto pero hay que tener en cuenta el tiempo dedicado al cambio de proceso
- Dos criterios posibles:
 - En el orden de entrada
 - En el orden de llegada a la cola (FIFO) que utilizaremos como *default* del Round Robin

Algoritmos de planificación

Round-Robin, turno rotatorio

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

**Round-Robin, turno
rotatorio**

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

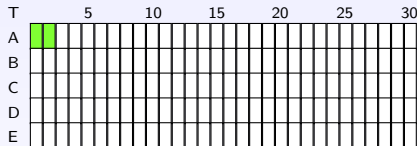
Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Se ejecuta



█ Ejecución
 █ Listo
 █ Espera E/S
 █ Sin cargar
 █ Terminado

Algoritmos de planificación

Round-Robin, turno rotatorio

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

**Round-Robin, turno
rotatorio**

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

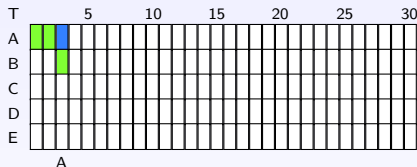
Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

A listo
Se ejecuta



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Algoritmos de planificación

Round-Robin, turno rotatorio

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

**Round-Robin, turno
rotatorio**

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

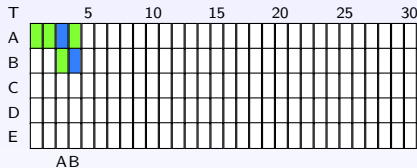
Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Se ejecuta
A listo



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Algoritmos de planificación

Round-Robin, turno rotatorio

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

**Round-Robin, turno
rotatorio**

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

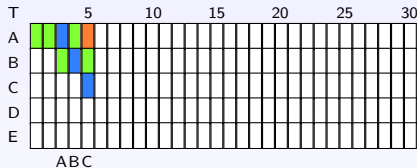
Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

E/S se bloquea
Se ejecuta
A listo



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Algoritmos de planificación

Round-Robin, turno rotatorio

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

**Round-Robin, turno
rotatorio**

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

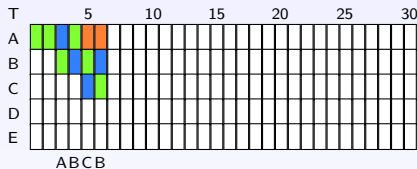
Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

A listo
Se ejecuta



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Algoritmos de planificación

Round-Robin, turno rotatorio

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

**Round-Robin, turno
rotatorio**

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

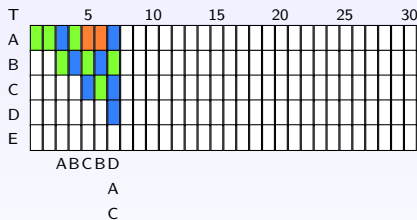
Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

A listo
Se ejecuta
A listo
A listo



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Algoritmos de planificación

Round-Robin, turno rotatorio

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

**Round-Robin, turno
rotatorio**

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

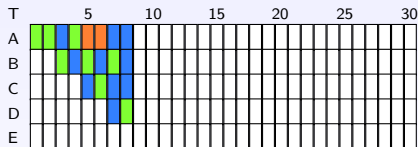
Planificación en
UNIX

Evaluación de
algoritmos

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

A listo

Se ejecuta



ABCBDA

AC

CB

■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Algoritmos de planificación

Round-Robin, turno rotatorio

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

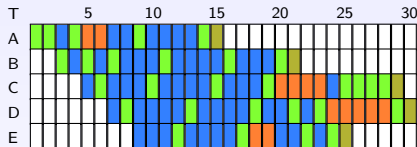
HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2



ABCBDACBEDACBEDCBDEDE C

ACBEDACBEDCBDE

CBEDACBEDCB

DACBED

■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

① Uso de CPU = 100 % (29/29)

② Rendimiento = 5/29 (proc/q)

③ Tiempo de retorno (medio) = $(14 + 18 + 24 + 23 + 16)/5 = 19q$

④ Tiempo de espera/respuesta (medio) = $(7 + 13 + 13 + 14 + 10)/5 = 11,4q$

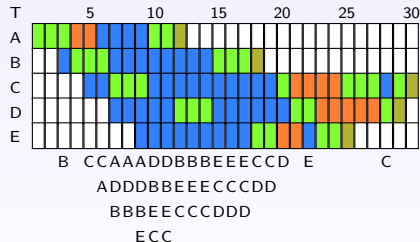
Algoritmos de planificación

Round-Robin, turno rotatorio

Planificación de
procesos

$q=3$

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2



Ejecución Listo Espera E/S Sin cargar Terminado

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Algoritmos de planificación

Round-Robin, turno rotatorio

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Parámetro crítico de diseño: longitud del cuanto

- Si es muy pequeño los procesos cortos pasan rápidamente, pero hay sobrecarga del procesador (gestión interrupciones de reloj, planificación, expedición)
- Si es muy grande degenera en FCFS
- Referencia: debe ser algo mayor que el tiempo necesario para una interacción normal
- Efectivo en sistemas de carácter general, de tiempo compartido o procesos de transacciones
- Favorece procesos con carga de procesador frente a procesos con carga de E/S (éstos no aprovechan el cuanto)

Algoritmos de planificación

Primero el proceso más corto (SPN)

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS
Round-Robin, turno
rotatorio

**Primero el proceso
más corto (SPN)**

Menor tiempo
restante (SRT)
HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

- Suele tener una política apropiativa (no expulsiva).
- Se selecciona el proceso con menor tiempo esperado de ejecución. ¿? (previsión de tiempo esperado)
- Un proceso corto saltará a la cabeza de la cola, sobrepasando a trabajos largos.
- Se reduce la previsibilidad de los procesos largos.
- Si la estimación de tiempo del proceso no es correcta, el sistema puede abandonar el trabajo.
- Posibilidad de inanición para los procesos largos.

Algoritmos de planificación

Primero el proceso más corto (SPN)

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS
Round-Robin, turno
rotatorio

**Primero el proceso
más corto (SPN)**
Menor tiempo
restante (SRT)
HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Estimaciones

Trabajos por lotes o repetitivos

Estimación del programador o estadísticas en función de los tiempos de ejecución pasados.

Procesos interactivos

En lugar de tiempo de trabajo, tiempo de cada ráfaga (se supone que siguen una distribución uniforme). La estimación se calcula en función de ráfagas pasadas (media o con alfa). $S_{n+1} = \alpha t_n + (1 - \alpha)S_n$ con $0 < \alpha < 1$

- S_1 : valor pronosticado (no calculado). Puede eliminarse en sucesivos cálculos o sustituirse por T_1
- Si $\alpha \rightarrow 1$ se reflejan rápidamente los cambios, pero si son efectos aislados desestabilizan la media más tiempo.
- Conviene dar más peso a los valores más recientes

Algoritmos de planificación

Primero el proceso más corto (SPN)

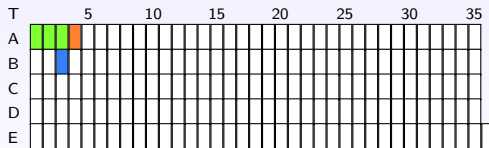
Planificación de
procesos

Proceso por lotes

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Suma 5

Suma 6



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

**Primero el proceso
más corto (SPN)**

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Algoritmos de planificación

Primero el proceso más corto (SPN)

Planificación de
procesos

Proceso por lotes

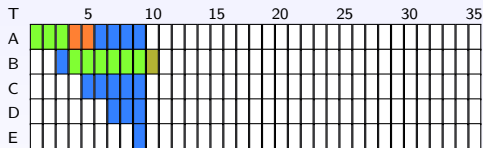
Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Suma 5

Suma 8

Suma 6

Suma 4



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

**Primero el proceso
más corto (SPN)**

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Algoritmos de planificación

Primero el proceso más corto (SPN)

Planificación de
procesos

Proceso por lotes

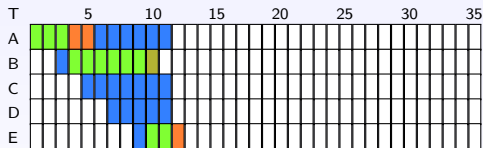
Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Suma 5

Suma 8

Suma 6

Suma 4



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

**Primero el proceso
más corto (SPN)**

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

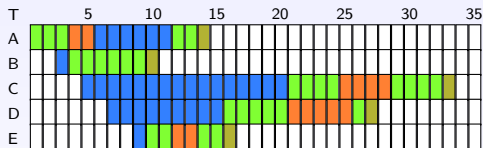
Algoritmos de planificación

Primero el proceso más corto (SPN)

Planificación de
procesos

Proceso por lotes

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

1. Uso de CPU = $29/32$
2. Rendimiento = $5/32$ (proc/q)
3. Tiempo de retorno (medio) = $(13 + 7 + 28 + 20 + 7)/5 = 15q$
4. Tiempo de espera/respuesta (medio) = $(6 + 1 + 16 + 9 + 1)/5 = 6,6q$

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Algoritmos de planificación

Primero el proceso más corto (SPN)

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS
Round-Robin, turno
rotatorio

**Primero el proceso
más corto (SPN)**

Menor tiempo
restante (SRT)
HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Proceso interactivo

Hay múltiples formas para estimar tiempo inicial de un proceso nuevo, por ejemplo:

- No sabemos nada sobre las ráfagas: $S_1 = 0$ (los procesos nuevos son preferentes)
- Tomando $S_1 = T_1$ (problema: hay que saber el T_1)
- Tomando $S_1 = cte$ (ej: media de las ráfagas de procesos interactivos anteriores en el sistema)

Algoritmos de planificación

Primero el proceso más corto (SPN)

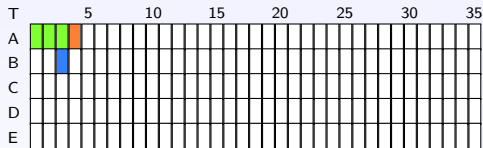
Planificación de
procesos

Proceso interactivo

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Previsto 2

Previsto 6



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS
Round-Robin, turno
rotatorio

**Primero el proceso
más corto (SPN)**

Menor tiempo
restante (SRT)
HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Algoritmos de planificación

Primero el proceso más corto (SPN)

Planificación de
procesos

Proceso interactivo

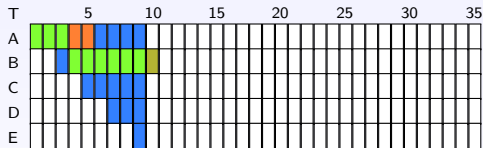
Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Previsto 2

Previsto 4

Previsto 5

Previsto 2



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Algoritmos de planificación

Primero el proceso más corto (SPN)

Planificación de
procesos

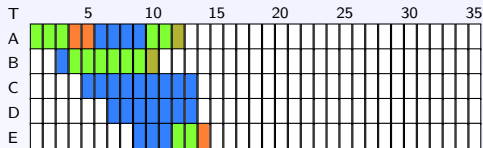
Proceso interactivo

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Previsto 4

Previsto 5

Previsto 2



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Algoritmos de planificación

Primero el proceso más corto (SPN)

Planificación de
procesos

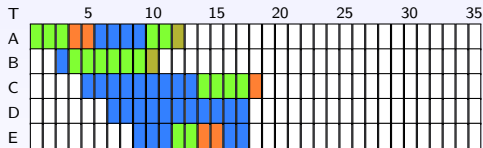
Proceso interactivo

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Previsto 4

Previsto 5

Previsto 2



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS
Round-Robin, turno
rotatorio

**Primero el proceso
más corto (SPN)**

Menor tiempo
restante (SRT)
HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Algoritmos de planificación

Primero el proceso más corto (SPN)

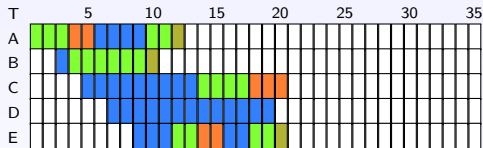
Planificación de
procesos

Proceso interactivo

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Previsto 4

Previsto 5



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS
Round-Robin, turno
rotatorio

**Primero el proceso
más corto (SPN)**

Menor tiempo
restante (SRT)
HRRN

Otras políticas

Planificación en
UNIX

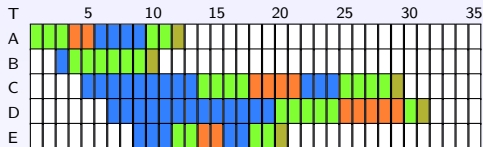
Evaluación de
algoritmos

Algoritmos de planificación

Primero el proceso más corto (SPN)

Proceso interactivo

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

1. Uso de CPU = $28/29$
2. Rendimiento = $5/29$ (proc/q)
3. Tiempo de retorno (medio) = $(11 + 7 + 24 + 24 + 11)/5 = 15,4q$
4. Tiempo de espera/respuesta (medio) = $(4 + 1 + 12 + 13 + 5)/5 = 7q$

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Algoritmos de planificación

Primero el proceso más corto (SPN)

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

**Primero el proceso
más corto (SPN)**

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Características

- Mejora el rendimiento global: tiempo de retorno y tiempo de espera/respuesta
- Es posible la inanición para los procesos largos
- No es conveniente para tiempo compartido o procesamiento de transacciones (por ser apropiativa)
- Se reduce la previsibilidad de los procesos largos (pueden variar mucho con pequeños cambios en las condiciones)

Algoritmos de planificación

Menor tiempo restante (SRT)

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

**Menor tiempo
restante (SRT)**

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

- Es una versión preferente de la política de primero el proceso más corto.
- Debe estimar el tiempo de proceso.

Función de selección

Mínimo tiempo restante de ejecución ($t. \text{ total} - t. \text{ consumido}$)

Modo de decisión

Preferente (no apropiativa)

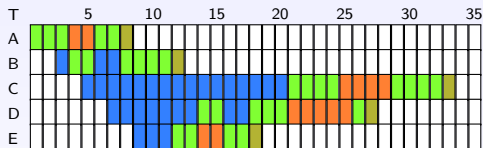
Algoritmos de planificación

Menor tiempo restante (SRT)

Planificación de
procesos

Proceso por lotes

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

1. Uso de CPU = $29/32$
2. Rendimiento = $5/32$ (proc/q)
3. Tiempo de retorno (medio) = $(7 + 9 + 28 + 20 + 9)/5 = 14,6q$
4. Tiempo de espera/respuesta (medio) = $(0 + 3 + 16 + 9 + 3)/5 = 6,2q$

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

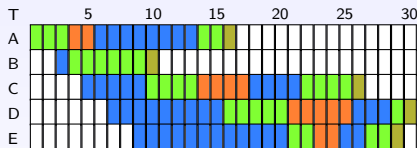
Algoritmos de planificación

Menor tiempo restante (SRT)

Planificación de
procesos

Proceso interactivo para $S_1 = 3$

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

1. Uso de CPU = $29/29$
2. Rendimiento = $5/29$ (proc/q)
3. Tiempo de retorno (medio) = $(7 + 9 + 28 + 20 + 9)/5 = 14,6q$
4. Tiempo de espera/respuesta (medio) = $(0 + 3 + 16 + 9 + 3)/5 = 6,2q$

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos



Algoritmos de planificación

Menor tiempo restante (SRT)

Planificación de procesos

Introducción

Tipos de planificación

Políticas de planificación

Algoritmos de planificación

FCFS

Round-Robin, turno rotatorio

Primero el proceso más corto (SPN)

Menor tiempo restante (SRT)

HRRN

Otras políticas

Planificación en UNIX

Evaluación de algoritmos

SRT favorece a los procesos cortos

Ventaja

No genera interrupciones adicionales (vs. Round Robin)

Desventaja

Debe contabilizar los tiempos de servicio transcurridos \Rightarrow sobrecarga



Algoritmos de planificación HRRN

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

Elige el proceso con la tasa más alta. Donde la tasa es

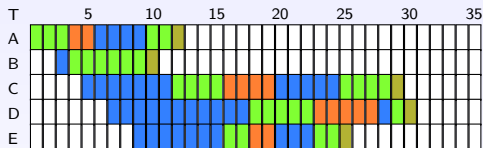
$$tasa = \frac{\text{tiempo consumido esperando al procesador} + \text{tiempo de servicio esperado}}{\text{tiempo de servicio esperado}}$$

- Procesos cortos \Rightarrow denominador pequeño \Rightarrow tasa de respuesta alta
- Envejecimiento sin servicio \Rightarrow denominador grande \Rightarrow tasa de respuesta alta \Rightarrow los procesos pueden competir con los cortos

Algoritmos de planificación HRRN

Planificación de
procesos

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

1. Uso de CPU = $29/29$
2. Rendimiento = $5/29$ (proc/q)
3. Tiempo de retorno (medio) = $(11 + 7 + 24 + 23 + 16)/5 = 16,2q$
4. Tiempo de espera/respuesta (medio) = $(4 + 1 + 12 + 12 + 10)/5 = 7,8q$

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

FCFS

Round-Robin, turno
rotatorio

Primero el proceso
más corto (SPN)

Menor tiempo
restante (SRT)

HRRN

Otras políticas

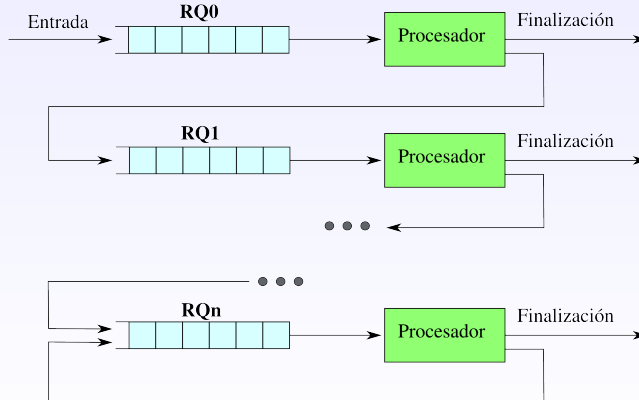
Planificación en
UNIX

Evaluación de
algoritmos

Otras políticas

Realimentación multinivel

- Penaliza a los trabajos que han estado ejecutándose durante más tiempo.
- No se conoce el tiempo de ejecución restante del proceso.
- Política FIFO no apropiativo (FIFO preferente)



Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

**Realimentación
multinivel**

Por reparto
equitativo

Planificación
garantizada

Planificación en
UNIX

Evaluación de
algoritmos

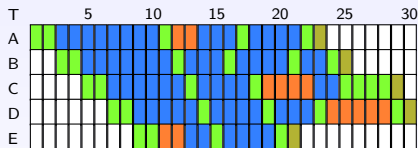
Otras políticas

Realimentación multinivel

Planificación de
procesos

Con 5 colas de prioridad

Proceso	Llegada	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2



■ Ejecución
 ■ Listo
 ■ Espera E/S
 ■ Sin cargar
 ■ Terminado

1. Uso de CPU = $29/29$
2. Rendimiento = $5/29$ (proc/q)
3. Tiempo de retorno (medio) = $(22 + 22 + 24 + 23 + 12)/5 = 20,6q$
4. Tiempo de espera/respuesta (medio) = $(15 + 16 + 12 + 12 + 6)/5 = 12,5q$

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

Realimentación
multinivel

Por reparto
equitativo

Planificación
garantizada

Planificación en
UNIX

Evaluación de
algoritmos

Otras políticas

Realimentación multinivel

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

**Realimentación
multinivel**

Por reparto
equitativo

Planificación
garantizada

Planificación en
UNIX

Evaluación de
algoritmos

Problema

Los **procesos largos**: llevados gradualmente hacia abajo. Problema: **pueden sufrir inanición** en colas de prioridad baja si llegan muchos procesos cortos continuamente

Soluciones

- Cuanta menor es la prioridad se pueden asignar más cuantos de tiempo de ejecución
- Tras cierto tiempo de espera en cola, se le cambia a una cola de prioridad mayor



- $P_i = Base_i + \frac{CPU_i}{2} + \frac{GCPU_i}{4W_k}$

	Proceso A			Proceso B			Proceso C		
	Prioridad	Proceso	Grupo	Prioridad	Proceso	Grupo	Prioridad	Proceso	Grupo
0	60	0	0	60	0	0	60	0	0
	1	1							
	2	2							
	:	:							
	60	60							
1	90	30	30	60	0	0	60	0	0
				1	1				1
				2	2				2
				:	:				:
				60	60				60
2	74	15	15	90	30	30	75	0	30
	16	16							
	17	17							
	:	:							
	75	75							
3	96	37	37	74	15	15	67	0	15
						16		1	16
						17		2	17
						:		:	:
						75		60	75
4	78	18	18	81	7	37	93	30	37
	19	19							
	20	20							
	:	:							
	78	78							
5	98	39	39	70	3	18	76	15	18

Grupo 1
Grupo 2



Otras políticas

Planificación garantizada

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

Realimentación

multinivel

Por reparto

equitativo

Planificación

garantizada

Planificación en
UNIX

Evaluación de
algoritmos

- A cada proceso se le garantiza un uso equitativo de la CPU ($1/n$, siendo n el número de procesos en espera de ser ejecutados).
- Cada vez que un proceso va a ser asignado tiempo de CPU se comprueba la relación **tiempo real/tiempo prometido** de todos los procesos y se adjudica la CPU a aquel proceso que tiene el ratio más pequeño.



Planificación en UNIX

Características

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

Planificación en
UNIX

Características

Proceso

Prioridad

Planificación clásica

Evaluación de
algoritmos

- Emplea realimentación multinivel usando turno rotatorio en cada una de las colas de prioridad.
- La prioridad de cada proceso se calcula cada segundo.
- La prioridad base divide los procesos en bandas fijas de prioridad.
- Se utiliza un factor de ajuste para impedir que un proceso salga fuera de la banda que tiene asignada.

Planificación en UNIX

Proceso

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

Planificación en
UNIX

Características

Proceso

Prioridad

Planificación clásica

Evaluación de
algoritmos

- Cada segundo (1s) el planificador recalcula las prioridades de los procesos y los organiza en niveles de prioridad en función de dichos valores.
- Cada décima de segundo (0.1 s) el planificador selecciona el proceso que tenga máxima prioridad y le asigna tiempo de CPU.
- Si el proceso termina su cuanto de ejecución (no hay bloqueo), el proceso pasa a la cola de su nivel de prioridad.
- Si el proceso se bloquea durante su cuanto, el planificador selecciona inmediatamente otro proceso y le asigna tiempo de CPU.
- Si un proceso retorna de una llamada al sistema y hay un proceso listo con mayor prioridad, el proceso de menor prioridad es desalojado de la CPU.
- Cada 4 centésimas de segundo (0.04 s) el planificador recalcula la prioridad del proceso que está usando tiempo de CPU.

Planificación en UNIX

Prioridad

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

Planificación en
UNIX

Características

Proceso

Prioridad

Planificación clásica

Evaluación de
algoritmos

- La prioridad de un proceso se calcula con la fórmula siguiente:

$$Pri = \frac{C_1}{\text{Uso reciente de CPU}} + \frac{C_2}{\text{Prioridad estática (nice)}}$$

- Consecuencias:
 - La prioridad de los procesos disminuye si utilizan mucho tiempo de CPU en una ventana de tiempo determinada.
 - Por el contrario, procesos con mucha demanda de E/S tenderán a tener prioridades altas.
 - Los procesos con un valor de prioridad estática (nice) alto, tendrán menor prioridad.
- Prioridad por bandas. En orden decreciente de prioridad:
 - Intercambio.
 - Control de dispositivos de E/S de bloques.
 - Gestión de archivos.
 - Control de dispositivos de E/S de caracteres.
 - Procesos de usuario.

Planificación en UNIX

Planificación clásica

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

Planificación en
UNIX

Características

Proceso

Prioridad

Planificación clásica

Evaluación de
algoritmos

- $CPU_i = \frac{CPU_{i-1}}{2}$
- $P_i = Base_i + \frac{CPU_i}{2} + nice_i$

	Proceso A		Proceso B		Proceso C	
	Prioridad Proceso		Prioridad Proceso		Prioridad Proceso	
0	60	0	60	0	60	0
	1					
	2					
	3					
	4					
	5					
1	75	30	60	0	60	0
			1			
			2			
			3			
			4			
			5			
2	67	15	75	30	60	0
					1	
					2	
					3	
					4	
					5	
3	63	7	67	15	75	30
		8				
		9				
		10				
		11				
		12				
4	78	33	63	7	67	15
			8			
			9			
			10			
			11			
			12			
5	68	16	76	33	63	7

Evaluación de algoritmos

Planificación de
procesos

Introducción

Tipos de
planificación

Políticas de
planificación

Algoritmos de
planificación

Otras políticas

Planificación en
UNIX

Evaluación de
algoritmos

- Seleccionar criterio de optimización
- Métodos de evaluación:
 - Modelado Determinista:
 - Medida (números exactos) de la carga de CPU proceso.
 - Utilidad académica o en sistemas que ejecutan los mismos programas.
 - Modelo de colas:
 - Estimación (probabilística) de la carga de CPU
 - Requisitos de E/S de los sistemas tiempos de llegada de procesos.
 - Permite comparar los distintos algoritmos.
 - Problema: Arbitrariedad del modelo matemático derivado de cada algoritmo.
 - Simulaciones:
 - Medida de la respuesta de los algoritmos a secuencias generadas aleatoriamente, mediante distribuciones o eventos grabados

Parte II

Planificación multiprocesador

Clasificación de sistemas

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

- Débilmente acoplado, procesador distribuido, o cluster
 - Cada procesador tiene su propia memoria y canales de E/S
- Procesadores de funcionalidad especializada
 - Ej: procesadores de E/S
 - Controlados por un procesador maestro.
- **Multiprocesamiento fuertemente acoplado**
 - Los procesadores comparten memoria principal
 - Bajo el control integrado del sistema operativo.



Granularidad

Tipos

Planificación de
procesos

Clasificación de
sistemas

Granularidad
Tipos

Comparativa

Planificación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Tipos de granularidad de sincronización o frecuencia de sincronización entre los procesos del sistema:

- Paralelismo independiente
- Grano grueso
- Grano medio
- Grano fino

Granularidad

Tipos

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Tipos

Comparativa

Planificación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Paralelismo independiente

- Aplicaciones o trabajos separados.
- Sin sincronización entre los procesos.
- Ej: sistemas de tiempo compartido.

Grano grueso

- Sincronización entre procesos a muy alto nivel, con poca frecuencia.
- Se trata como procesos concurrentes que ejecutan en un monoprocesador con multiprogramación.
- Se le puede dar soporte en multiprocesador con pocos cambios en software de usuario.

Granularidad

Tipos

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Tipos

Comparativa

Planificación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Grano medio

- Necesidades de comunicación e interacción mayores.
- Aplicación = colección de hilos.
- Los hilos interactúan entre sí y se coordinan con frecuencia.

Grano fino

- Aplicaciones altamente paralelas.
- Área muy especializada y fragmentada, con muchas propuestas diferentes.

Granularidad Comparativa

Planificación de
procesos

Clasificación de
sistemas

Granularidad
Tipos
Comparativa

Planificación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Tamaño del grano	Descripción	Intervalo de sincronización (instrucciones)
Fino	Paralelismo inherente en un único flujo de instrucciones	< 20
Medio	Procesamiento paralelo o multitarea dentro de una única aplicación	20-200
Grueso	Multiprocesamiento de procesos concurrentes en un entorno multiprogramado	200-2000
Muy grueso	Procesamiento distribuido entre nodos de una red para conformar un único entorno de computación	2000-1M
Independiente	Múltiples procesos no relacionados	(N/D)



Planificación

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Vinculación
Asignación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Aspectos de diseño

- Vinculación y asignación de procesos a procesadores
- Multiprogramación en procesadores individuales
- Ejecutar cada proceso (dispatching)

Planificación

Vinculación de procesos

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Vinculación
Asignación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Dinámica

Tratar procesos como recursos colectivos y asignar procesos a procesadores bajo demanda:

- Una cola global de procesos
- Se asigna el procesador que esté libre

Estática

Asignar procesos permanentemente a un procesador:

- Permite implementar planificación por grupos (más adelante)
- Cola a corto plazo para cada procesador.
- Menos sobrecarga en la función de planificación.
- Un procesador puede estar desocupado mientras otro procesador tiene trabajo acumulado.



Planificación

Vinculación de procesos

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación
Vinculación
Asignación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Balance dinámico de carga

- Cada procesador tiene su cola
- Los hilos se mueven de una cola a otra
- Ej: Linux

Planificación

Asignación de procesos a procesadores

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Vinculación
Asignación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

- Enfoque maestro/esclavo
- Enfoque de colegas (o camaradas)
- Enfoques mixtos

Planificación

Asignación de procesos a procesadores

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Vinculación
Asignación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Enfoque maestro/esclavo

- Las funciones claves del núcleo se ejecutan siempre en el mismo procesador (maestro).
- El resto de los procesadores ejecutan programas de usuario (esclavos).
- El maestro es responsable de la planificación.
- Cada esclavo envía pedidos de servicios al maestro. Por ejemplo, llamada de E/S

Desventajas

- Un fallo del maestro hace que falle todo el sistema.
- El maestro se puede convertir en un cuello de botella.

Planificación

Asignación de procesos a procesadores

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Vinculación
Asignación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Enfoque de colegas

- El núcleo del sistema operativo puede ejecutar en cualquier procesador.
- Cada procesador ejecuta su propio planificador y elige proceso de entre los disponibles.
- El sistema operativo es más complicado:
 - Tiene que asegurarse que dos procesadores no elijan el mismo proceso.
 - Tiene que asegurarse de que los procesos no se extravían
 - Tiene que incorporar técnicas para resolver y sincronizar competencia por recursos

Planificación

Asignación de procesos a procesadores

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Vinculación

Asignación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Enfoques mixtos

- Subconjunto de procesadores para procesar el núcleo
- Gestionar necesidades entre procesos del núcleo y otros procesos sobre:
 - Prioridad
 - Historia de ejecución

En sistemas multiprocesador tradicionales

Procesos

- Única cola para todos los procesos o múltiples colas para administrar prioridades.
- Todas las colas alimentan el único colectivo de procesadores.

Hilos

- Hilo = Unidad de ejecución, separada del resto de la definición del proceso.
- Aplicación implementada como **conjunto de hilos** que cooperan y ejecutan de forma concurrente en el mismo espacio de direcciones.
- **Paralelismo real** en una aplicación: hilos de la aplicación son **ejecutados simultáneamente** en procesadores separados.



Planificación Asignación

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Asignación

Compartición de
carga

Por grupos

Procesador dedicado

Planificación
dinámica

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Cuatro enfoques distintos:

- Compartición de carga.
- Planificación por grupos (o en pandilla).
- Asignación de procesador dedicado.
- Planificación dinámica.

Planificación

Compartición de carga

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Asignación

Compartición de
carga

Por grupos

Procesador dedicado

Planificación
dinámica

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

- Los procesos no se asignan a procesador particular
- Cola global de hilos listos.
 - Todos los hilos listos de todas las aplicaciones están en una o varias colas globales
- Carga distribuida uniformemente entre los procesadores:
 - No hay procesadores ociosos mientras haya trabajo
 - Un procesador ocioso selecciona hilo de cola
- No se precisa planificador centralizado:
 - En el procesador disponible se ejecuta la rutina de planificación del sistema operativo para seleccionar el siguiente hilo a ejecutar.

Planificación

Compartición de carga

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Asignación

Compartición de
carga

Por grupos

Procesador dedicado

Planificación
dinámica

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Desventajas de la compartición de carga

- Acceso a colas globales: necesita exclusión mutua
 - Hay un cuello de botella si hay muchos procesadores buscando trabajo al mismo tiempo.
- Los hilos expulsados pueden retomar la ejecución en un procesador distinto al de su última ejecución
 - El uso de la caché de cada procesador es menos eficaz.
- Los procesos con alto grado de coordinación entre sus hilos pueden verse desfavorecidos:
 - Los hilos en varios procesadores necesitan coordinarse \Rightarrow los cambios de proceso pueden comprometer el rendimiento global

Planificación

Por grupos

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Asignación
Compartición de
carga
Por grupos
Procesador dedicado
Planificación
dinámica

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Se planifica un conjunto de hilos afines para su ejecución en un conjunto de procesadores al mismo tiempo. Relación uno a uno

- Reducción de bloqueos por sincronización (no esperarán mucho, al estarse ejecutando)
- Necesidad de menos cambios por proceso
- Útil para aplicaciones paralelas en general, y específicamente para grano medio/fino: el rendimiento se degrada cuando hay partes listas y partes que no se ejecutan.

¿Dónde se utiliza la planificación por grupos?

- Planificación simultanea de hilos de un proceso.
- Aplicaciones paralelas cuyo rendimiento es sensible a la coordinación estrecha de sus hilos.

Planificación

Por grupos

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Asignación
Compartición de
carga

Por grupos
Procesador dedicado
Planificación
dinámica

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Planificación uniforme por número de grupos vs. ponderada por el número de hilos de cada proceso

Dos procesos, uno con 4 hilos y otro con un hilo.

División uniforme

	Grupo 1	Grupo 2
Pr. 1		
Pr. 2		Desocupado
Pr. 3		Desocupado
Pr. 4		Desocupado

Desperdicio 37.5%

División por pesos

	Grupo 1	Grupo 2	
Pr. 1			Pr. 1
Pr. 2		Des.	Pr. 2
Pr. 3		Des.	Pr. 3
Pr. 4		Des.	Pr. 4

Desperdicio 15%



Planificación

Procesador dedicado

Planificación de procesos

Clasificación de sistemas

Granularidad

Planificación

Planificación

Asignación
Compartición de carga
Por grupos
Procesador dedicado
Planificación dinámica

Planificación de Linux

Planificación en UNIX SVR4

Planificación en Windows

Enfoque opuesto al reparto de carga

- A cada proceso se le asigna un número de procesadores igual al número de sus hilos
 - Cada procesador está dedicado al hilo hasta que la aplicación concluye (monoprogramación)
- Cuando el programa finaliza, los procesadores retornan a la reserva general para posibles asignaciones a otros programas.

Planificación

Procesador dedicado

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Asignación
Compartición de
carga
Por grupos
Procesador dedicado
Planificación
dinámica

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Inconvenientes

- Pueden quedar ociosos algunos procesadores
- No hay multiprogramación de los procesadores (entre hilos de distintos procesos).

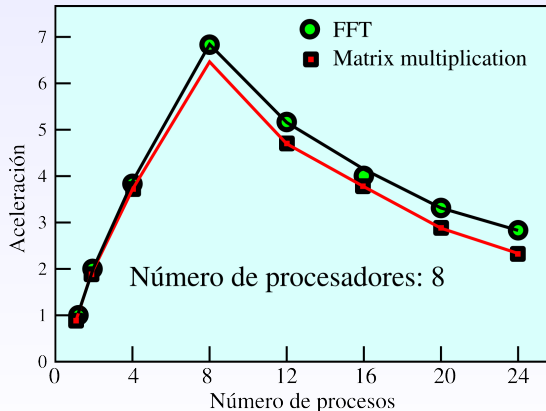
Ventajas

- En sistemas altamente paralelos (decenas o centenas de procesadores):
 - Cada procesador representa una pequeña fracción del coste del sistema
 - La utilización del procesador deja de ser una medida importante de la eficacia o rendimiento (tengo muchos)
- Evitar totalmente cambios de procesos durante la vida de un programa supone una mejora sustancial en su velocidad:
 - El tiempo se emplea en sincronización

Planificación

Procesador dedicado

- ¿Cuántos procesadores asignamos a un programa en un momento dado?
 - Conjunto residente de actividad
 - Limitar el número de hilos activos al número de procesadores



Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Asignación

Compartición de
carga

Por grupos

Procesador dedicado

Planificación
dinámica

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows



Planificación

Procesador dedicado

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Asignación
Compartición de
carga
Por grupos
Procesador dedicado
Planificación
dinámica

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Conjunto residente de actividad

Mínimo número de actividades (hilos) que deben ser planificados simultáneamente sobre procesadores para que la aplicación tenga un progreso aceptable.

Fragmentación del procesador

Quedan procesadores ociosos que no son suficientes en número, o que no están organizados adecuadamente para dar soporte a los requisitos de las aplicaciones pendientes.

Planificación

Planificación dinámica

El número de hilos de un proceso puede cambiar durante el curso de su ejecución.

Cuando trabajo pide uno o más procesadores

- Si hay procesadores ociosos: asignárselos
- Si no los hay y si es un trabajo nuevo, ubicarlo en un único procesador, quitándoselo al trabajo que tenga más de un procesador
- Si no se puede satisfacer parte de la solicitud (número de procesadores), se mantiene el trabajo pendiente hasta que:
 - Haya un procesador esté disponible
 - El trabajo rescinda la solicitud (ej: ya no necesita tantos procesadores)

Liberar procesadores

- Mirar si hay trabajos en la cola de listos.
- Elegir los que no tienen procesador asignado: asignarles un único procesador (ej: trabajos nuevos).

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Asignación
Compartición de
carga
Por grupos
Procesador dedicado

Planificación
dinámica

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Planificación de Linux

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Planificación de
Linux

Hilos no de tiempo
real

Planificación en
UNIX SVR4

Planificación en
Windows

Clases de planificación

- SCHED_FIFO: Hilos de tiempo real FIFO
- SCHED_RR: Hilos de tiempo real Round-Robin
- SCHED_OTHER: Otros hilos no de tiempo real

Dentro de cada clase pueden utilizarse múltiples prioridades

Linux 2.6 usa el planificador nuevo $O(1)$

- El tiempo que tarda en elegir el próximo proceso y asignarlo a un procesador es constante
 - Independiente de la carga del sistema.
 - Independiente del número de procesadores.

Planificación de Linux

Hilos no de tiempo real

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Planificación de
Linux

Hilos no de tiempo
real

Planificación en
UNIX SVR4

Planificación en
Windows

- Cada tarea no de tiempo real tiene prioridad entre 100 y 139 (default 120).
- Esta prioridad puede variar dinámicamente de acuerdo al tiempo que la tarea esté esperando por eventos E/S.
 - Se favorece las tareas limitadas por la E/S \Rightarrow mejora respuesta interactiva.
- Cuantos (rodajas) entre 10 y 200ms.
 - A mayor prioridad se le asigna un cuanto de tiempo más largo.

Planificación de Linux

Hilos no de tiempo real

Planificación de
procesos

Maneja dos tipos de colas, 1 bit por cada cola

Clasificación de
sistemas

Granularidad

Planificación

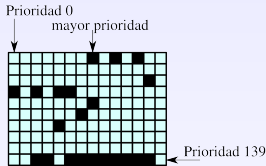
Planificación

Planificación de
Linux

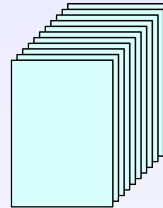
Hilos no de tiempo
real

Planificación en
UNIX SVR4

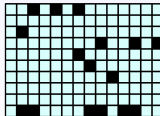
Planificación en
Windows



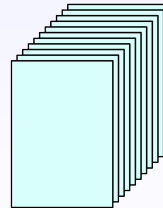
Array de 140 bits para las
colas activas



140 colas para los
procesos activos



Array de 140 bits para las
colas activas



140 colas para los
procesos activos

Planificación en UNIX SVR4

Características

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Características

Planificación en
Windows

- Planificador expulsivo de prioridades estáticas
- 160 niveles de prioridad divididos en tres clases
- Clases de prioridad:
 - Mayor preferencia: procesos de tiempo real
 - Siguiendo mayor preferencia: procesos en modo núcleo
 - Menor preferencia: otros procesos de usuario
- Inserción de puntos de expulsión seguros para el núcleo básico:
 - Región de código donde todas las estructuras de datos del núcleo están actualizadas y son consistentes o bien están bloqueadas por un semáforo.

Planificación en UNIX SVR4

Características

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Características

Planificación en
Windows

**Clase de
prioridad** **Valor
global** **Secuencia de
planificación**

Tiempo real	159 ⋮ 100	Primero ↓
Kernel	99 ⋮ 60	
Tiempo compartido	59 ⋮ 0	Último

Planificación en UNIX SVR4

Características

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Características

Planificación en
Windows

Tiempo Real (159 – 100)

- Serán elegidos para ejecutar antes que cualquier proceso de núcleo o de tiempo compartido
- Pueden expulsar procesos de núcleo o de usuario.

Núcleo (99 – 60)

Serán elegidos para ejecutar antes que cualquier proceso de tiempo compartido (pero no antes que tiempo real)

Tiempo Compartido (59-0)

Menor prioridad



Planificación en UNIX SVR4

Características

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Características

Planificación en
Windows

- Turnos rotatorios dentro de cada nivel
- En los procesos de T.C. el nivel puede variar:
 - Disminuye al utilizar todo el cuanto
 - Aumenta cuando se bloquea por recurso o evento
- El tamaño del cuanto depende del nivel:
 - 100 ms para prioridad 0 — 10 ms para prioridad 59

Planificación en Windows

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

**Planificación en
Windows**

Planificación con
prioridades
Ejemplo de
prioridades

- Planificador expulsivo (no apropiativo) con sistema flexible de niveles de prioridad
- Prioridades organizadas en dos clases:
 - Tiempo Real
 - Variable

Planificación en Windows

Planificación con prioridades

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Planificación con
prioridades
Ejemplo de
prioridades

Colas de
prioridad de
tiempo real

Mayor 31

•
•
•

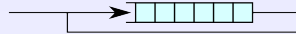
Menor 16

Clases de
prioridad
variable

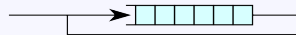
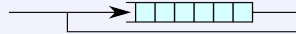
Mayor 15

•
•
•

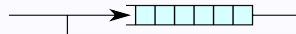
Menor 0



•
•
•



•
•
•



Planificación en Windows

Ejemplo de prioridades

Planificación de
procesos

Clasificación de
sistemas

Granularidad

Planificación

Planificación

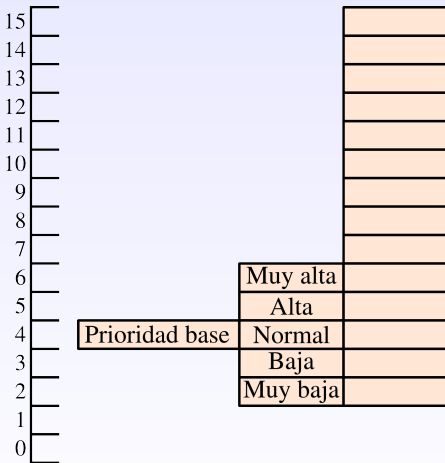
Planificación de
Linux

Planificación en
UNIX SVR4

Planificación en
Windows

Planificación con
prioridades

Ejemplo de
prioridades



Prioridad
proceso

Prioridad
hilo

Prioridad
dinámica