Sistemas de Computación

1er Semestre 2011

Scheduling de Procesos

Dr. Wenceslao Palma M.<wenceslao.palma@ucv.cl>

Generalmente en un sistema computacional existe un procesador real.

Cada proceso puede ser visto como un procesador virtual en donde se ejecuta un programa.

Es tarea del kernel asignar un tiempo de procesador a los procesos que se encuentran en el sistema.

Dado que el procesador es un recurso compartido es necesario asignarlo en forma “estratégica” lo cual se denomina scheduling de procesos. El componente del kernel que realiza esta tarea es el scheduler. También existe scheduling asociado a otro recurso, como el disco duro, pero en este caso al referirnos al scheduler consideramos el asociado al procesador.

El propósito del scheduling va en beneficio del tiempo de respuesta, productividad y eficiencia del procesador.

En muchos sistemas el scheduling se divide en tres funciones independientes, considerando la frecuencia relativa con la que se ejecutan, estas son; scheduling a largo, medio y corto plazo.

Además, cada función se relaciona con alguna cola involucrada en las transiciones de estado de un proceso.

Largo plazo

Decisión de añadir procesos al conjunto de procesos a ejecutar, es decir cuáles son los programas admitidos en el sistema. Esto permite el control del grado de multiprogramación.

Esta decisión es importante debido a que mientra más procesos se crean, menor es el porcentaje de tiempo asignado para que cada uno se ejecute.

Esto involucra, considerando el diagrama de transición de Unix, las transiciones 8 a 3 y 8 a 5.

La decisión se puede apoyar en un algoritmo del tipo FCFS (Fisrt-Come, Fisrt Served).

En un sistema de tiempo compartido, que sucede con una solicitud de login?

Medio plazo

Básicamente tiene que ver con la función de intercambio entre disco y memoria.

Esto involucra las transiciones 5 a 3 y 6 a 4.

Corto plazo

Es el de ejecución más frecuente ya que se relaciona con los procesos que se encuentran en estado 3.

El scheduling a corto plazo se activa cuando ocurre un suceso que puede conducir a la interrupción del proceso actual u ofrecer la oportunidad de expulsar del procesador al actual proceso en favor de otro.

Algoritmos de Planificación

Generalmente utilizan criterios de planificación mediante los cuales es posible evaluarlos.

Criterios orientados al usuario (rendimiento)

Tiempo de retorno: tiempo de ejecución real + tiempo de espera por los recursos.

Tiempo de respuesta: tiempo que transcurre entre la emisión de una solicitud hasta que se comienza a recibir la respuesta. Es mejor desde el punto de vista del usuario.

Plazos: plazos para el término de procesos, todo gira en torno a maximizar el porcentaje de plazos cumplidos.

Otro criterio

Previsibilidad: los procesos se deben ejecutar en el mismo tiempo y con el mismo costo sin importar la carga del sistema.

Criterios orientados al sistema (rendimiento)

Productividad: maximizar el número de procesos terminados por unidad de tiempo.

Uso del procesador: porcentaje de tiempo en el cual el procesador está ocupado.

Criterios orientados al sistema (otros criterios)

Equidad: igual trato para todos los procesos. Ningún proceso debe sufrir inanición.

Prioridades: si está presente el uso de prioridades se debe favorecer a los de mayor prioridad.

Equilibrio de recursos: se deben mantener ocupados los recursos del sistema. Se favorece a procesos que no utilizan recursos sobrecargados.

Existe dependencia entre los criterios, por lo cual es imposible garantizarlos en forma simultánea.

FCFS(First-come, First served)

Es la más simple ya que es FIFO. El proceso se incorpora a la cola de los procesos listos y será seleccionado cuando sea el más antiguo en la fila.

El tiempo de despacho depende del orden de llegada.

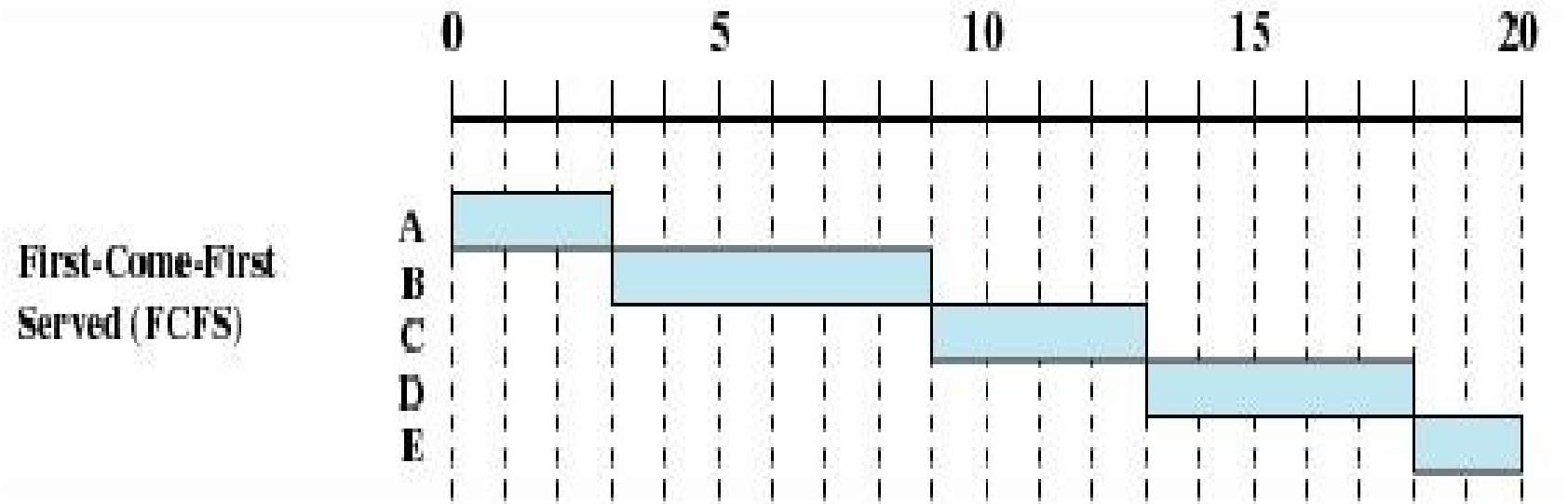
Los procesos intensivos en uso de CPU tiene preferencia antes que los intensivos en E/S.

FCFS tiene mal comportamiento en sistemas de tiempo compartido.

FCFS no es atractivo para un sistema monoprocesador y se plantea su utilización junto a un esquema de prioridades.

Consideremos el siguiente ejemplo de scheduling de procesos:

Proceso	A	B	C	D	E	Media
Llegada	0	2	4	6	8	
$T_{servicio}$ (T_s)	3	6	4	5	2	
$T_{finalización}$	3	9	13	18	20	
$T_{retorno}$ (T_r)	3	7	9	12	12	8.60
T_r/T_s	1.00	1.17	2.25	2.40	6.00	2.56



Round-Robin

Está diseñado especialmente para sistemas de tiempo compartido.

Cada proceso recibe una fracción de tiempo.

Periódicamente se genera una interrupción de reloj. El proceso que está en ejecución es enviado a la cola de procesos en estado ready y se selecciona el siguiente proceso usando FCFS.

El scheduler toma el primer proceso de la cola y programa un cronómetro para que interrumpa luego de un quantum de tiempo. Luego, pueden suceder dos cosas:

El proceso tiene una duración menor que el quantum, en cuyo caso se libera la CPU y el scheduler procede con el siguiente proceso de la cola ready.

El proceso tiene una duración mayor que el quantum asignado, en cuyo caso se genera un cambio de contexto y el proceso se ubicará al final de la cola ready.

En general el tiempo promedio de retorno es grande.

Un aspecto de diseño importante es el tamaño del quantum.

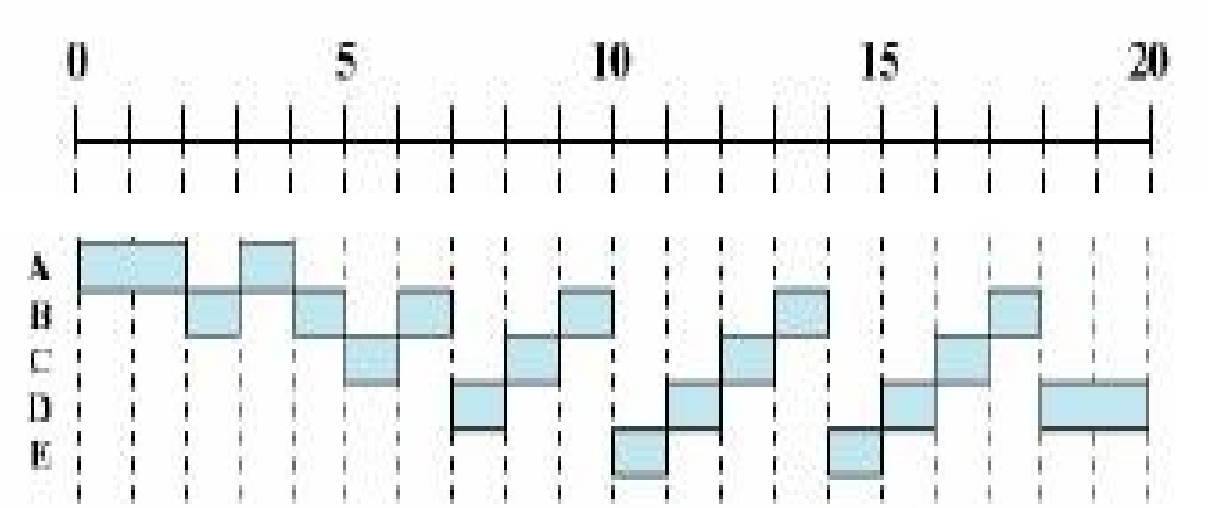
Si el quantum es muy grande Round-Robin degenera en FCFS. Por el contrario, el sobrecosto en cambios de contexto será alto.

Un quantum de tiempo debe ser “grande” respecto al tiempo que demora un cambio de contexto.

En el siguiente ejemplo el tamaño del quantum es 1:

Proceso	A	B	C	D	E	Media
Llegada	0	2	4	6	8	
$T_{servicio}$ (T_s)	3	6	4	5	2	
$T_{finalización}$	4	18	17	20	15	
$T_{retorno}$ (T_r)	4	16	13	14	7	10.80
T_r/T_s	1.33	2.67	3.25	2.80	3.50	2.71

Round-Robin (RR), $q = 1$



Shorter Process Next (SPN)

Es una forma de reducir el sesgo favorable al proceso más largo propio de FCFS.

Es una política no expulsiva donde si se atiende a los procesos con tiempo de procesamiento esperado más corto, el tiempo promedio de despacho será menor.

De esta forma, un proceso corto pasará al inicio de la cola por delante de los procesos más largos.

Lo difícil de esta estrategia es el conocimiento del tiempo exigido por cada proceso.

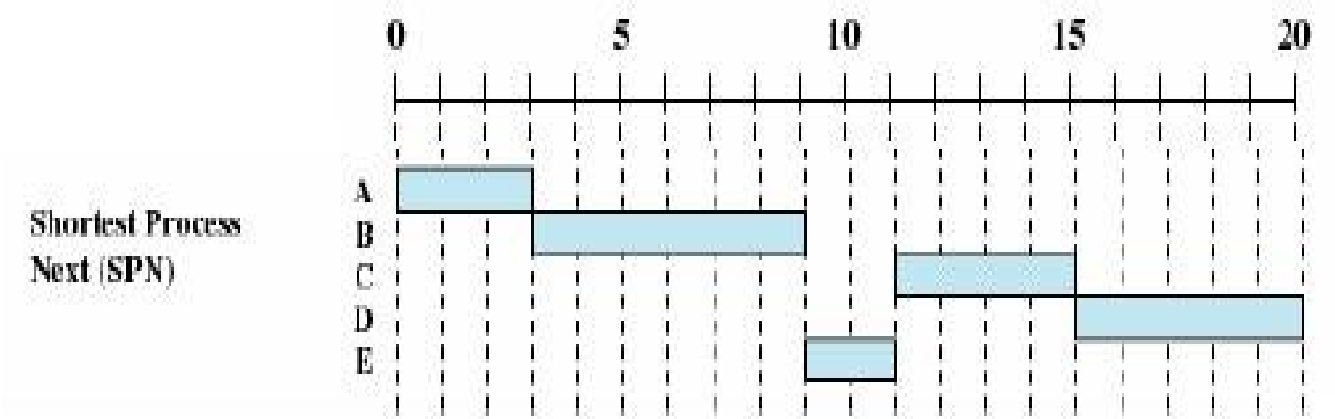
Generalmente se utiliza como predictor el promedio exponencial. Cuando hay procesos en la cola del estado ready se escoge aquél que tenga menor predictor.

$$S_{n+1} = \alpha T_n + (1-\alpha) S_n$$

$S_n + 1 : predicción$
$T : duración\ anterior$
$S_n : predicción\ anterior.$
$\alpha : determina\ el\ peso\ relativo\ de\ las\ observaciones$

Proceso	A	B	C	D	E	Media
Llegada	0	2	4	6	8	
$T_{servicio}$ (T_s)	3	6	4	5	2	
$T_{finalización}$	3	9	15	20	11	
$T_{retorno}$ (T_r)	3	7	11	14	3	7.60
T_r/T_s	1.00	1.17	2.75	2.80	1.50	1.84

Es posible que exista inanición para procesos largos mientras lleguen continuamente procesos cortos.



Shortest Remaining Time

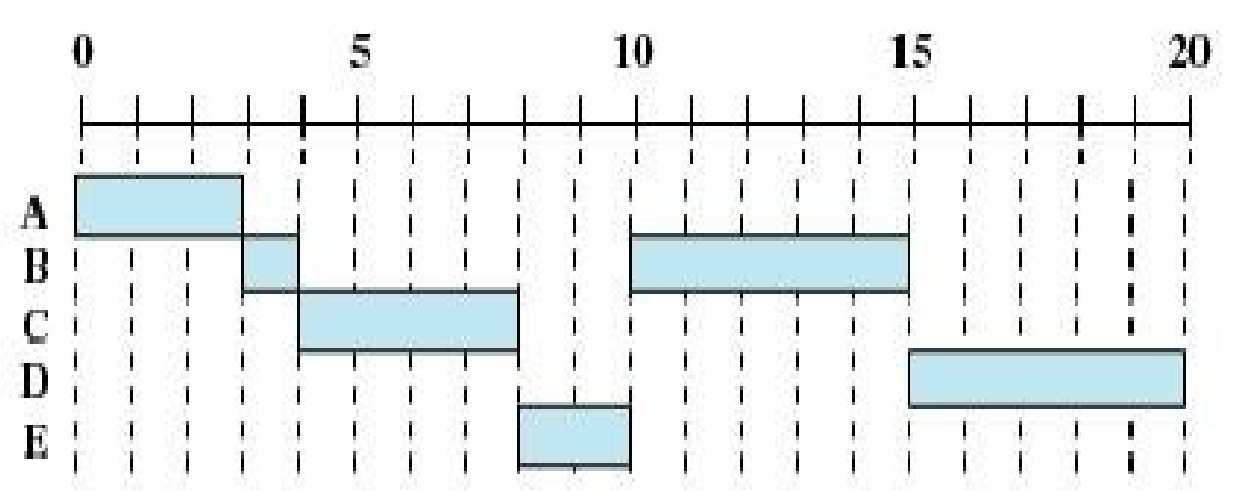
El scheduler siempre escoge al proceso que tiene menor tiempo restante. Es una versión expulsiva de SPN.

Cuando llega un nuevo proceso a la cola ready, puede quedarle un tiempo esperado de ejecución menor que al proceso que está ejecutándose. Por lo tanto el scheduler podría expulsar el proceso actual cuando llega un nuevo proceso.

El tiempo de retorno es menor que SPN ya que los trabajos cortos son atendidos en forma inmediata.

Proceso	A	B	C	D	E	Media
Llegada	0	2	4	6	8	
$T_{servicio}$ (T_s)	3	6	4	5	2	
$T_{finalización}$	3	15	8	20	10	
$T_{retorno}$ (T_r)	3	13	4	14	2	7.20
T_r/T_s	1.00	2.17	1.00	2.80	1.00	1.59

Shortest Remaining Time (SRT)



Retroalimentación

Si no se tiene información sobre el tiempo esperado de ejecución de los procesos no se puede emplear SPN ni SRT.

Una forma de dar preferencia a trabajos cortos es castigando a los procesos que estan en ejecución durante mucho tiempo. Por lo tanto, al no disponer de lo que pasará a futuro se utiliza el tiempo de ejecución consumido hasta el momento.

Para ello se utiliza un quantum de tiempo asignado a cada proceso + un mecanismo dinámico de asignación de prioridades.

Cuando un proceso ingresa al sistema es ubicado en la cola RQ0. Cuando retorna al estado ready se incorpora a RQ1. Después de cada ejecución siempre será enviado a un nivel con menor prioridad. Asi un proceso corto terminará en forma rápida.

Se favorece a los procesos cortos frente a los más viejos y largos. Dentro de cada cola se utiliza FCFS. Excepto en la cola de menor prioridad donde se usa Round-Robin.

Un proceso largo puede sufrir inanición.

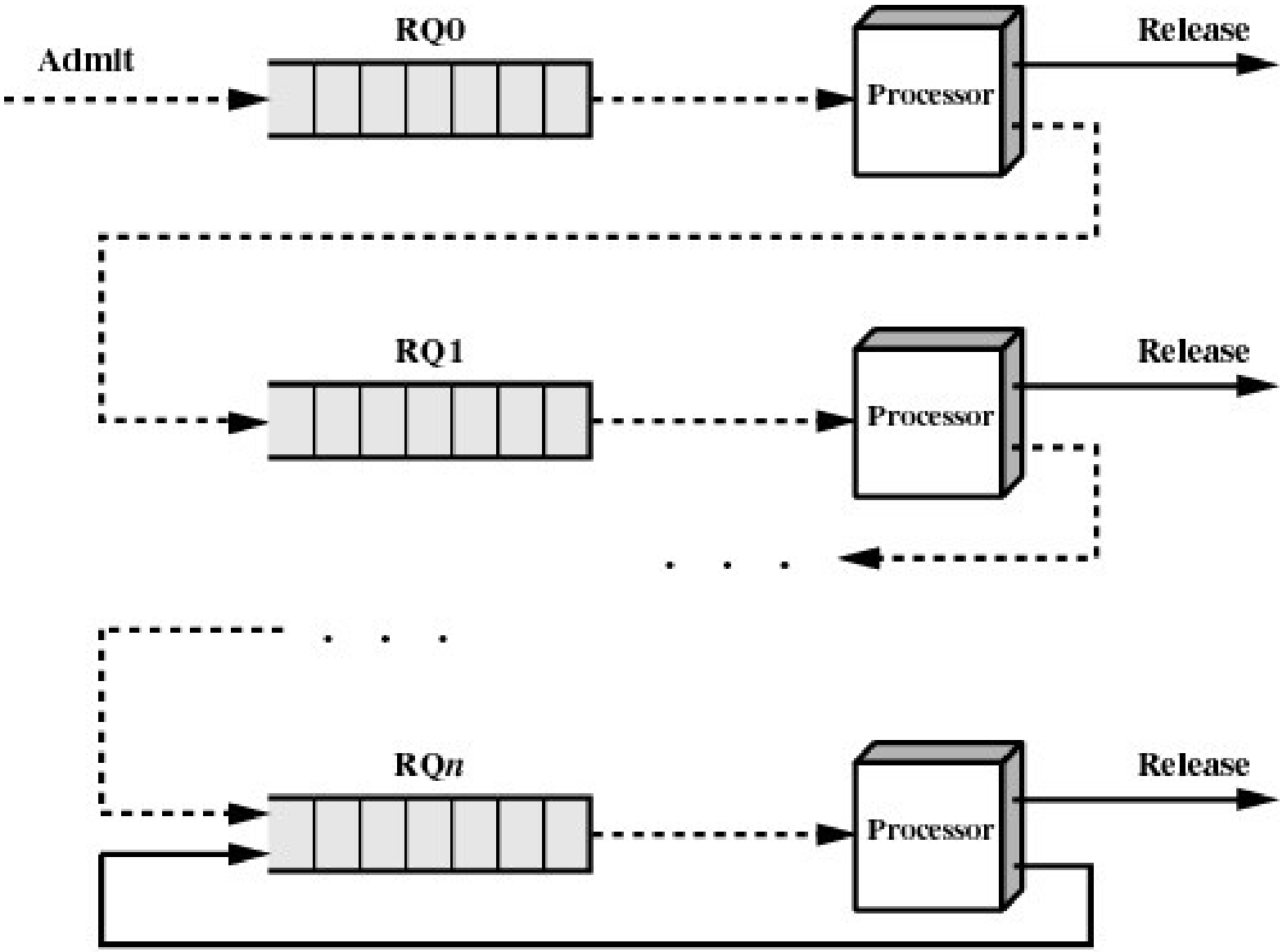


Figure 9.10 Feedback Scheduling

Considerando q=1

Proceso	A	B	C	D	E	Media
Llegada	0	2	4	6	8	
$T_{servicio}$ (T_s)	3	6	4	5	2	
$T_{finalización}$	4	20	16	19	11	
$T_{retorno}$ (T_r)	4	18	12	13	3	10.00
T_r/T_s	1.33	3.00	3.00	2.60	1.50	2.29