



Contenido

Contenido

concepto criterios algoritmos ejemplos

Tema 3. Admin. y planificación de procesos. 4h

3.1 Concepto de planificación

Criterios generales de planificación Tipos de planificadores

3.2 Criterios cuantitativos de planificación

3.3 Algoritmos de planificación

Clasificación: expropiativos/no expropiativos Políticas de planificación simples Políticas de planificación híbridas.

3.4 Ejemplos:

Linux y Windows XP



3.1 Concepto de planificación

Contenido

concepto criterios algoritmos ejemplos

3.1 Concepto de planificación

Dept. Arquitectura de Computadores

3 Univ. Málaga



3.1 Concepto de planificación conceptos básicos

Contenido

concepto criterios algoritmos ejemplos

- Procesos alternan fases de cómputo con esperas por E/S
- Clasificación de procesos
 - a) Intensivos en CPU (ráfagas de CPU largas)
 - b) Intensivos en E/S (ráfagas de CPU cortas)



ráfaga de CPU corta espera por E/S



3.1 Concepto de planificación

conceptos básicos

CPU

Run

Contenido

concepto criterios algoritmos ejemplos

- © Con sólo un proceso, la CPU permanecería ociosa en las esperas por entrada/salida
- Multiprogramación

Procesos

- Permite ejecutar varios procesos concurrentemente
- Aprovechar al máximo la CPU
- Función del planificador
 - decidir qué proceso accede a la CPU

Ready

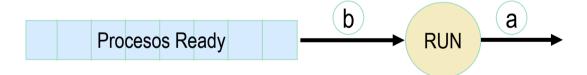


3.1 Concepto de planificación ¿qué es planificar?

Contenido

concepto criterios algoritmos ejemplos

- Una política de planificación responde a:
 - a) Cuándo y en qué condiciones expulsamos a un proceso en ejecución
 - Preemption (adelantamiento, expropiación)
 - b) Qué proceso de la cola listos despachamos
 - Scheduler (planificador)



- Resumiendo:
- La planificación son las políticas y mecanismos que deciden el orden en que se completan los procesos.



3.1 Concepto de planifiación criterios generales del planificador

Contenido

concepto criterios algoritmos ejemplos

Objetivo del planificador

 Maximizar las prestaciones del computador según ciertos criterios

Criterios generales

- Maximizar el rendimiento
 - Uso de la CPU, n° de trabajos terminados por u. tiempo,....
- Ser estadísticamente predecible
 - Posibilidad de estimar cuánto va a tardar un trabajo
- Ser imparcial
 - No debe discriminar injustamente a unos proc. sobre otros
- Aprovechar los recursos
 - Mantener en uso los distintos subsistemas del HW



3.1 Concepto de planifiación prioridad

Contenido

concepto criterios algoritmos ejemplos

Concepto de prioridad

- Derecho de preferencia de un proc. sobre otro
- Se materializa en un número

Origen de la prioridad de un proceso

- Externo al SO: Aspectos políticos
 - Departamet. u organización al que pertenece el propietario del proceso; Precio pagado por el tiempo de CPU; ...
- Interno al SO: criterios
 - Reducir el Tesp; asegurar tiempos de respuesta; aprovechar recursos; evitar discriminaciones;

Pueden ser

- Estáticas: fáciles de implementar. No adaptat.
- Dinámicas: Sensibles a cambios en el sist.
 - **Envejecimiento**: aumento de prioridad de un proceso que lleva mucho tiempo esperando.



3.1 Concepto de planifiación

tipos de planificadores

Contenido

concepto criterios algoritmos ejemplos

Tipos de planificadores

- Largo plazo
- Medio plazo

Corto plazo **DISCO SWAPPED** medio (TRANSFERIDOS) exit **BATCH READY RUN** largo corto (COLA DE (LISTOS) (EJECUCIÓN) **EJECUCIÓN**) Interactivos-**MEMORIA BINARIO BLOCK EJECUTABLE** (BLOQUEADOS)

Dept. Arquitectura de Computadores

Univ. Málaga



3.1 Concepto de planifiación tipos de planificadores

Contenido

concepto criterios algoritmos ejemplos

Largo plazo

- Poca frecuencia de ejecución
- Para procesos por lotes (batch o jobs)
- Lanza programas a ejecución (crea procesos)
- Objetivos
 - Mantener constante el grado de multiprogramación
 - Si acaba un proceso, lanza otro
 - Mantener una buena mezcla de procesos
 - Tantos procesos intensivos en CPU como intensivos en E/S
- No siempre existe un planificador de este tipo

Los interactivos pasan direct. a Ready

- Filtrado de otra naturaleza
 - Por el número de terminales disponible (limita los usuarios)
 - Usuarios que abandonan si el sistema está muy cargado



3.1 Concepto de planifiación tipos de planificadores

Contenido

concepto criterios algoritmos ejemplos

Medio plazo

- Puede controlar
 - La ocupación de memoria (si hay poca, descarga procesos)
 - La carga del sistema (si es mucha, suspende procesos)
 - Puede mejorar la mezcla de procesos
- No es relevante en SO modernos con M. Virt.
 - Los procesos que no referencian sus páginas, las van perdiendo porque otros procesos las reemplazan

Corto plazo

- Mantiene ocupada la CPU
- Se invoca muy frecuentemente
 - Por tanto no debe consumir mucho tiempo de CPU
 - Ejemplo: Quantum de 10ms. Planificador consume 1ms
 - 9% del tiempo de CPU ocupado por el planificador



3.2 Criterios de planificación

Contenido

concepto criterios algoritmos ejemplos

3.2 Criterios de planificación



3.2 Criterios de planificación criterios cuantitativos

Contenido

concepto criterios algoritmos ejemplos

- El planificador debe optimizar valores objetivos y cuantitativos = medidas de rendimiento:
 - Utilización de la CPU: 0%-100%
 - Porcentaje de tiempo con la CPU ocupada
 - Descontar el tiempo de CPU consumido por el SO
 - Capacidad de procesamiento, rendimiento o productividad (throughput): trabajos/hora, procesos/seq.
 - Trabajos terminados por unidad de tiempo
 - Tiempo de retorno (t. real o elapsed): Tr
 - Tiempo desde que lanzas un programa hasta que termina.
 - Tiempo de espera: Te=Tr-Tproc
 - Parte de Tret en que el proceso está esperando en Ready
 - Tiempo de respuesta
 - En sist. multitarea mide la interactividad: desde que pides algo al SO hasta que éste responde.

planificación

3.2 Criterios de planificación ejemplo y optimización de criterios

Contenido

concepto criterios algoritmos ejemplos

- @ Ejemplo: time ls -lR /usr/X11
 - El comando time devuelve inf. temp. del ls

3.38 user, 13.82 sys, 21.24 elapsed, 80%

- Tr: desde que empieza hasta que termina 21.24s
- Tproceso: 3.38 + 13.82 = 17.2
 - 3.38s en ejecución del proceso (modo user)
 - 13.82s en ejec. de syscalls llamadas por el proceso (núcleo)
- Te = 21.24 (3.38 + 13.82) = 4.04s. 17.2/21.24 x 100= 80% (suponiendo que no hay bloqueos por E/S)

Optimizar criterios

- Maximizar la utilización y productividad
- Minimizar los tiempos (retorno, espera, resp.)
- En general se max. o min. valores medios
- A veces se optimizan valores max. o min.
 - Ejemplo: minimizar el valor max. del tiempo de respuesta
- Otras veces interesa optimizar la varianza



3.2 Criterios de planificación diferentes entornos = diferentes objetivos

Contenido

concepto criterios algoritmos ejemplos

- → Política de planificación
 - Conjunto de estrategias y decisiones tomadas para diseñar la planificación y conseguir los objetivos propuestos
- Sistemas por lotes
 - Capacidad de procesamiento/rendimiento
 - Tiempo de retorno
 - Utilización de la CPU
- Sistemas interactivos
 - Tiempos de respuesta (varianza)
 - Proporcionalidad complejidad/tiempo respuesta
- Sistemas en tiempo real
 - Cumplir tiempos límite de servicio
 - Predicibilidad o consistencia



Contenido

concepto criterios algoritmos ejemplos

3.3 Algoritmos de planificación



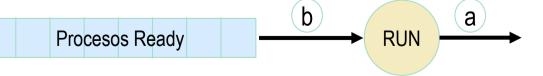
3.3 Algoritmos de planificación tipos de planificación

Contenido

concepto criterios algoritmos ejemplos

En la política de planificación se decide

- a) Preemption: cuándo expulsar a un proceso run
 - a) Cuando un proceso termina
 - b) Cuando el proceso inicia una operación E/S bloqueante (fin ráfaga CPU)
 - c) Cuando el proceso ya lleva cierto tiempo en run Tiempo máximo en Run: Quantum o rodaja de tiempo: una interrupción programada cada cierto tiempo lanza el planificador
 - d) Cuando aparece un proc. ready más prioritario (nuevo o procedente de un bloqueo)
- b) Scheduler: cómo ordenar la cola ready (=qué proceso de la cola listos despachamos)
 - a) Por orden de llegada (FIFO)
 - b) Por tiempo de ejecución del proceso o ráfaga de CPU
 - c) Por prioridad





3.3 Algoritmos de planificación planificación apropiativa/no apropiativa

Contenido

concepto criterios algoritmos ejemplos

Tipos de 'preemption'

- No expropiativa (no preemptive, sin expulsión)
 - Sólo considera las situaciones de preemption a) y b)
 - Implicaciones: mientras un proceso no termine o solicite una operación de E/S no puede despacharse otro a Run
- Expropiativa (preemptive, con expulsión)
 - También considera las situaciones de preemption c) y/o d)
 - **Implicaciones**
 - Un proceso puede expropiar a otro menos prioritario
 - Un proceso que agote su quatum tiene que dejar paso
 - Ventajas
 - Mejora la interactividad y baja Trespt. de proc. prioritarios
 - Se evita que un proceso acapare la CPU
 - Más seguro: un proc. colgado en Run no bloquea todo el SO
 - Inconveniente: aumenta el n° de cambios de contexto



3.3 Algoritmos de planificación algoritmos de planificación simples

Contenido

concepto criterios algoritmos ejemplos

- Planificadores de largo plazo
 - FIFO y SJF (Shortest Job First)
- Planificadores de corto plazo
 - No expropiativos
 - FCFS y Prioridad no expropiativo
 - Expropiativos
 - Round Robin, Prioridad expropiativo y SRT

Planifi-	a) Pre	emptio	n	b)	Sched	ule	Tipo	
cador	Nunca	E/S	Quantum	Prior.	FIFO	Tiempo	Prior.	Προ	
FIFO	Х				Χ			Largo	
SJF	Χ					Х		Largo Plazo	
FCFS		Χ			Χ				
Round Robin		Χ	Х		Х				
Pri. No Exprop		Χ					Χ	Corto Plazo	
Pri. Exprop		Χ		Χ			Χ	1 1020	
SRT		Χ		Χ		X			

planificación

3.3 Algoritmos de planificación planificación para procesos batch

Contenido

concepto criterios algoritmos ejemplos

FIFO: Monoproceso

- Ordenación por orden de llegada
- Ventaja: implementación simple
- Inconveniente: Tespera y Tretorno elevados
 - Ejemplo: En 0 llegan P1 (10s), P2 (2s) y P3 (1s)

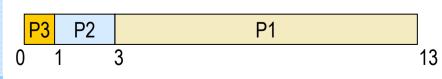


Te =
$$(0+10+12) / 3 = 7.33$$

Tr = $(10+12+13) / 3 = 11.66$

SJF (Shortest Job First): Monoproceso

- Ordenación por tiempo de ejecución más corto
- Ventaja: reduce Tespera y Tretorno
- Inconveniente: discrimina procesos largos



Te =
$$(0+1+3) / 3 = 1.33$$

Tr = $(1+3+13) / 3 = 5.66$

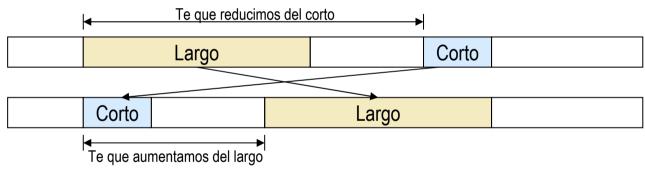


Contenido

concepto criterios algoritmos ejemplos

© Características de SJF

- Inf. temporal extraída de los parámetros del trabajo batch (proporcionada por el usuario)
- Proporciona un Te mínimo
 - Si adelantas un proceso corto a uno largo, se reduce más el Te del proc. corto de lo que aumenta el Te del proc. largo.



@ Ejemplo:

	Tproc	H _{Ilega}	H_{entra}	H _{salida}	Te	Tr
P1	6	0				
P2	3	2				
P3	2	3				
P4	1	7				



planificación de corto plazo

Contenido

concepto criterios algoritmos ejemplos

- FCFS (First Come First Served)
 - Preemption por E/S (No expropiativo)
 - Planificación por orden de llegada
 - Inconvenientes: sistema multiprogramado
 - Como en FIFO elevados Te y Tr
 - Efecto convoy: bajo índice de ocupación de recursos (i.e.los procesos limitados por E/S no pueden continuar, y bloquearse, hasta que terminen los anteriores, que pueden ser intensivos de CPU)
 - Ejemplo
 - Ahora planificas ráfagas de CPU. Ejemplo P1(en solitario)

	CPU (4s)	E/S	CPU(6s)
(4	5

	T _{rafaga}	H_{Ilega}	H _{entra}	H _{salida}	Te	Tr
P1	4,6	0,5				
P2	3	2				
P3	2	3				
P4	1	7				
		Medios:				



planificación de corto plazo

Contenido

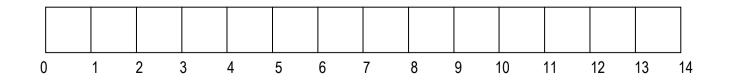
concepto criterios algoritmos ejemplos

RR (Round Robin)

- Preemption por E/S y por Quantum (Q)
- Planificación por orden de llegada
- Inconvenientes:
 - Aumenta el número de cambios de contexto
- Ventaja: sistema multitarea
 - Aumenta la interactividad

Q=1

Q=1	T _{rafaga}	H _{Ilega}	H _{entra}	H _{salida}	Te	Tr				
P1	6	0								
P2	3	2								
Р3	2	3								
P4	1	7								
	Valores Medios:									





3.3 Algoritmos de planificación planificación de corto plazo

Contenido

concepto criterios algoritmos ejemplos

RR (Round Robin)

- Si el grado de multiprogramación es N
 - Cada proceso recibe 1/N del tiempo de CPU
 - Trespuesta ≤ N x Q
 - No garantiza un Tresp muy pequeño para un proceso, pero si minimiza la desviación de Tresp y establece una cota superior
 - Propiedad muy útil en sistemas interactivos: un retardo esporádico no es grave y los usuarios tienen sensación de interactividad
- Decidir tamaño de O
 - Si Q el sistema degenera en FCFS

 - Relación de Q con el rendimiento de caché
 - Si Q♠ cuando un proceso entra no encuentra sus bloques cache
 - Si Q
 ✓ un proc. que trae sus bloques, no le da tiempo a aprovecharlos
 - Regla empírica: elegir Q de forma que:
 - El 80% de las ráfagas de CPU sean menores que Q



3.3 Algoritmos de planificación planificación de corto plazo

Contenido

concepto criterios algoritmos ejemplos

Planificación por prioridades

- Se asigna un valor numérico (prioridad) a cada proceso
- Se concede la CPU al proceso de mayor prioridad
 - normalmente menor valor ≡ mayor prioridad
 - expropiativa o no expropiativa
- SJF es un caso particular donde la prioridad es el tiempo previsto de ejecución (si se utiliza el tiempo previsto de la ráfaga para la planificación y hay preemption por E/S)
- Problema: inanición o postergación indefinida
 - puede que procesos de baja prioridad nunca alcancen la CPU
- Solución: envejecimiento (aging)
 - incrementar la prioridad de los procesos con el avance del tiempo



planificación de corto plazo

Contenido

concepto criterios algoritmos ejemplos

Prioridad No Expropiativo

- Preemption por E/S (No expropiativo)
- Planificación por prioridad
- Ejemplo: Prioridad más alta = 31

	T _{rafaga}	Prioridad	H _{Ilega}	H _{entra}	H _{salida}	Te	Tr
P1	5	4	0				
P2	3	3	2				
P3	4	5	3				
P4	10	3	5				
P5	6	4	7				

Dept. Arquitectura de Computadores

 SJF es un caso particular de Prioridad No Expropiativo con Prioridad=1/Tproceso



planificación de corto plazo

Contenido

concepto criterios algoritmos ejemplos

Prioridad Expropiativo

- Preemption por E/S y por prioridad
- Planificación por prioridad
- Ejemplo: Prioridad más alta = 31

	T _{rafaga}	Prioridad	H _{llega}	H _{entra}	H _{salida}	Te	Tr
P1	5	4	0				
P2	3	3	2				
P3	4	5	3				
P4	10	3	5				
P5	6	4	7				

- Ahora un proceso puede adelantar a otro con menos prioridad aunque éste esté en ejecución
- Útil en sistemas Tiempo Real (blando)



3.3 Algoritmos de planificación planificación de corto plazo

Contenido

concepto criterios algoritmos ejemplos

- SRT (Shortest Remaining Time first)
 - Preemption por E/S y por Prioridad
 - Planificación por prioridad=1/Trafaga
 - Inconvenientes:
 - Discrimina procesos largos: starvation o inanición
 - Ventaja:
 - Disminuye Te y aumenta productividad

	T _{rafaga}	H_{llega}	H _{entra}	H _{salida}	Te	Tr		
P1	6	0						
P2	3	2						
Р3	2	3						
P4	1	7						
Valores Medios:								

- Solución para el starvation: aging
 - Aumentar la prioridad de procesos que lleven mucho tiempo esperando (envejecimiento)

3.3 Algoritmos de planificación SRT y HRRN

Contenido

concepto criterios algoritmos ejemplos

Estimación del tiempo de ráfaga

Promedio exponencial

Estimación ráfaga n+1 Tiempo ráfaga n Estimación ráfaga n
$$\tau_{n+1}' = \alpha \cdot t_n' + (1-\alpha) \cdot \tau_n \qquad 0 \leq \alpha \leq 1$$

$$\tau_{n+1} = \alpha \cdot t_n + (1 - \alpha)\alpha \cdot t_{n-1} + \dots + (1 - \alpha)^j \alpha \cdot t_{n-j} + \dots + (1 - \alpha)^{n+1} \cdot \tau_0$$

- Es decir, el segundo término contiene un histórico
- Cuanto más antigua es la ráfaga menos pesa su duración
- Si α=0 no consideras el último tiempo de ráfaga
- Si α=1 no consideras la historia acumulada
- Típicamente α =1/2

Ejemplo de planificador con aging

- HRRN (Highest Response Ratio Next)
- Relación de respuesta RR=(Tesp+Traf)/Traf
- El proceso con mayor RR es el de más prioridad



resumen planificadores simples

Contenido

concepto criterios algoritmos ejemplos

Atendiendo a:

- Decisión de preemption:
 - Expropiativo o no expropiativo
- Decisión de scheduling:
 - Orden de llegada, tiempo de proc./ráfaga, prioridad

Preem. Sched.	No Expropiat.	Expropiat.
Orden de Ilegada	FIFO FCFS	Round Robin
Tiempo	SJF	SRT
Prioridad	Prio. No Exp.	Prio. Exp.

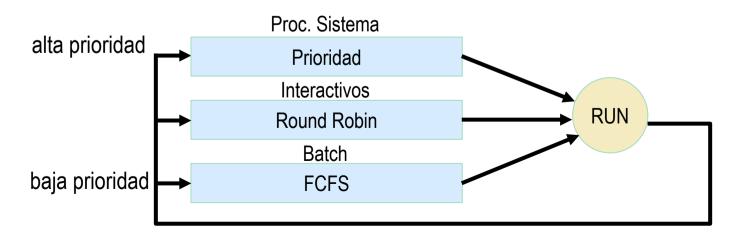


3.3 Algoritmos de planificación planificadores híbridos

Contenido

concepto criterios algoritmos ejemplos

- Los sitemas reales son heterogéneos
- Varias colas de listos (categorías y políticas específicas)
 - se clasifican los procesos: primer plano (foreground), segundo plano (background), sistema, ... cada grupo tiene requerimientos distintos
 - cada cola tiene asociado su propio algoritmo de planificación
 - el procesador se reparte entre los distintos niveles:
 - Las colas se planifican por prioridad. Problema: inanición o postergación indefinida
 - La CPU se reparte entre colas: sistema 70%, interactivos 20%....





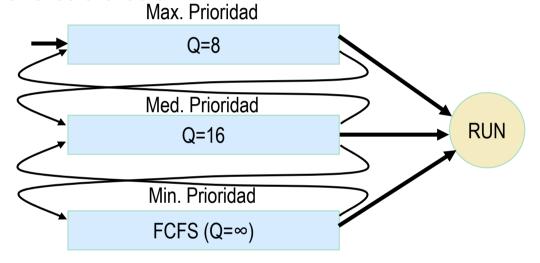
3.3 Algoritmos de planificación planificadores híbridos

Contenido

concepto criterios algoritmos ejemplos

Colas multinivel con realimentación

 Los procesos se clasifican solos automáticamente: no necesitamos clasificar a priori, permite a los procesos migrar de una cola a otra



- Los procesos entran por la cola Max. Prior.
 - Si agotan el Q de la cola actual bajan de cola; si no lo agotan, suben.
 - Ej. planificación de colas: Prioridad Expropiativa
 - Un proc. de Med. Prioridad no ejecuta (o es expropiado) mientras haya un proceso ready de Max. Prioridad.
 - Un proceso de Min. Prio. ejecuta cuando en las otras colas no hay procesos (están bloqueados por E/S)



3.3 Algoritmos de planificación planificadores híbridos

Contenido

concepto criterios algoritmos ejemplos

Colas multinivel con realimentación

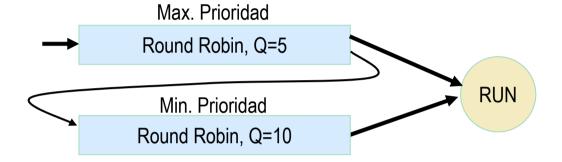
- Resultado: Clasificación automática
 - Los procesos intensivos en E/S se quedan arriba
 - Los procesos intensivos en CPU caen a la última cola
- Subir el Q cuando baja la prioridad proporciona una realimentación negativa al sistema
 - Si Q fuese cte, todos los proc. estarían en las colas extremas
- Inconveniente:
 - Un proceso que puntualmente tiene unas ráfagas de CPU largas puede acabar en la cola de mínima prioridad
 - Alternativas
 - Un proceso tiene que consumir su Q, "n" veces para bajar, y "n" puede crecer al bajar de cola.
 - Un proceso también puede subir (no solo bajar) en el sistema de colas:
 - » Si no consume un Q umbral "m" veces, se sube a la cola de arriba



3.3 Algoritmos de planificación planificadores híbridos

Contenido

concepto criterios algoritmos ejemplos @ Ejemplo: Colas multinivel con realiment.



	T _{rafaga}	H _{llega}	H _{entra}	H _{salida}	Te	Tr		
P1	8	0						
P2	4,2	2,10						
Р3	6	4						
P4	3,3	7,18		1				
P5	2	9						
Valores Medios:								

		1		1				
		1		1				
		1		1				
		1		1				



3.4 Ejemplos

Contenido

concepto criterios algoritmos ejemplos

3.4 Ejemplos



3.4 Ejemplos

planificación Linux

Contenido

concepto criterios algoritmos ejemplos

- Objetivos Unix (¿incompatibles?)
 - bajos tiempos de respuesta, buena productividad (trhoughput) para segundo plano, evitar inanición de procesos, conciliar las necesidades de procesos de alta y baja prioridad...
- Política de planificación
 - Planifica procesos
 - Planificación expropiativa en base a prioridades (dinámicas para procesos normales – estáticas TR)
 - Tiempo compartido mediante cuantos de tiempo
 - Procesos en tiempo real: tienen mayor prioridad y planificación diferenciada
 - Favorece procesos interactivos incrementando su prioridad



planificación Linux

Contenido

concepto criterios algoritmos ejemplos

Colas multinivel

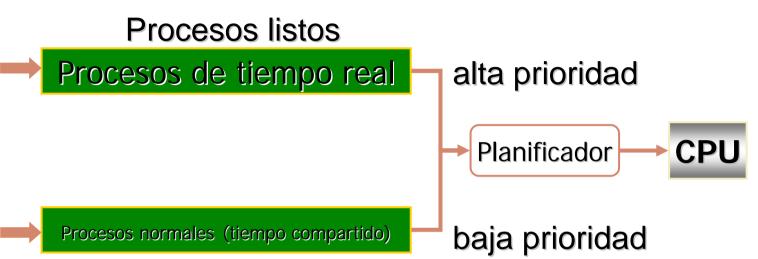
- Procesos de tiempo real (sin realimentación)
- Procesos normales (con realimentación)
- Os clases de procesos dos políticas
 - Procesos de tiempo real
 - Siempre se despachan antes que los procesos normales
 - Prioridades estáticas
 - No equitativo, lo importante: la prioridad absoluta
 - Procesos normales (tiempo compartido)
 - Prioridades dinámicas
 - Intenta ser equitativo
 - Procesos de baja prioridad pueden llegar a alcanzar la CPU
 - Favorecer interactivos



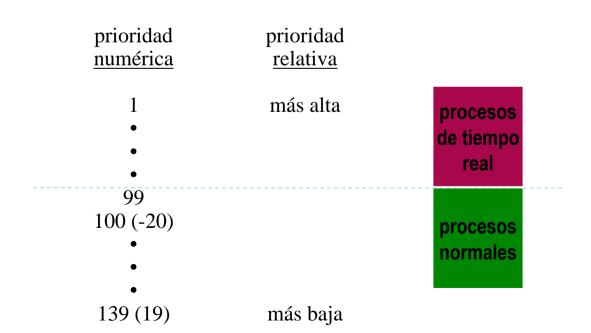
planificación Linux

Contenido

concepto criterios algoritmos ejemplos



Dept. Arquitectura de Computadores



Univ. Málaga

38



planificación Linux: tiempo compartido

Contenido

concepto criterios algoritmos ejemplos

- Planificación por prioridad expropiativa
 - Siempre se escoge el proceso de prioridad más alta
- Prioridad dinámica
 - Calculada a partir de la prioridad base (100..139)
 - Procesos heredan prioridad del padre
 - Usuarios pueden cambiar prioridad base (llamadas nice() y setpriority())
- Esquema de turno rotatorio (RR) para procesos de la misma prioridad
 - Cuanto fijo dependiente directamente de la prioridad estática del proceso



planificación Linux: tiempo compartido

Contenido

concepto criterios algoritmos ejemplos

Cálculo de la prioridad dinámica

- Cada proceso tiene un prioridad base (nice) asignada en la creación del proceso
- La dinámica se calcula a partir de la base para favorecer procesos interactivos
 - pdinámica = max (100, min(pestática bonus +5), 139))
 - bonus = 0..10
 - bonificación (hasta -5) a procesos intensivos en E/S
 - penalización (hasta +5) a procesos intensivos en CPU
- La bonificación se calcula en base al tiempo medio dormido (TMD en milisegundos)
 - $0 \le TMD \le 1000 \text{ ms}$
 - bonus = floor(TMD / 100)



planificación Linux: tiempo compartido

Contenido

concepto criterios algoritmos ejemplos Cálculo de la prioridad dinámica

- Intervalo "dinámico" (±5)
 - suficientemente grande para favorecer procesos interactivos
 - no suficiente para distorsionar totalmente prioridad base
 - Proceso con prioridad 12 nunca más urgente que prioridad 1
- La prioridad se recalcula cuando el proceso ha agotado su cuanto
- "Cuanto" fijo y dependiente de la prioridad estática del proceso (140 prioridad estática) x 20 si prioridad estática < 120 cuanto = (140 prioridad estántica) x 5 si prioridad estática ≥ 120

Descripción	Prioridad est.	Valor nice	Cuanto
Máxima	100	-20	800 ms
Alta	110	-10	600 ms
Defecto	120	0	100 ms
Baja	130	10	50 ms
Mínima	139	19	5 ms



planificación Linux: tiempo compartido

Contenido

concepto criterios algoritmos ejemplos

Turno rotatorio (RR) modificado

- Dos colas de procesos "listos"
 - Cola "listos activos" formada por los procesos que aún no han consumido su cuanto
 - Cola "listos expirados" formada por los procesos que ya han agotado su cuanto
- El proceso siguiente a ejecutar se elige sólo de los "activos"
- Si un proceso se bloquea sin consumir su cuanto, cuando se desbloquee vuelve a "listos activos"
- Si un proceso consume su rodaja, pasa a "listos expirados"

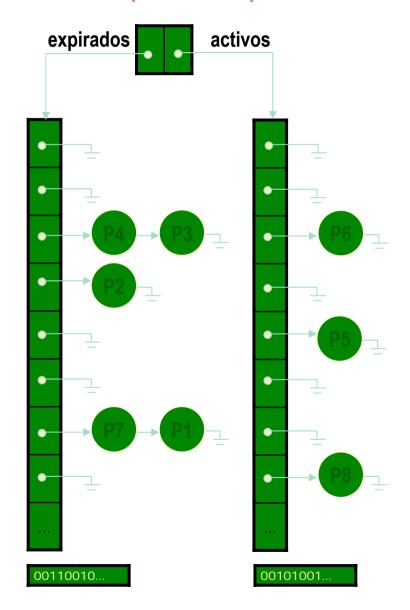


3.4 Ejemplos
planificación Linux: tiempo compartido

Contenido

concepto criterios algoritmos ejemplos Turno rotatorio (RR) modificado

- Evita inanición:
 - Un proceso menos prioritario se ejecutará aunque existan procesos más prioritarios listos (pero que hayan consumido su cuanto)
- Aparecen "épocas de ejecución"
 - Todos los procesos entran a ejecutarse en una época
 - Una época termina cuando todos los procesos han entrado a ejecución (no hay listos activos)
- Cambio de época
 - Basta cambiar cola de expirados por cola de activos





planificación Linux: tiempo compartido

Contenido

concepto criterios algoritmos ejemplos

Turno rotatorio (RR) modificado

- Implementación real un poco más complicada
- Clasificación de los procesos en Batch e Interactivos
- Interactive si:
 - prioridad dinámica ≤ 3 x prioridad estática / 4 + 28
 - depende del bonus (TMD) del proceso
- Procesos Batch
 - Cuando agota su cuanto es movido a "expirados"
- Procesos Interactivos
 - Cuando agota su cuanto normalmente permanece en "activos"
 - Es movido a expirados si:
 - El proceso expirado más viejo lleva mucho tiempo esperando
 - Hay un proceso expirado con mayor prioridad estática que el interactivo



planificación Linux: tiempo real

Contenido

concepto criterios algoritmos ejemplos

- Tiempo real se panifica antes que los procesos normales
- Planificación por prioridad expropiativa
 - Siempre se escoge el proceso TR de prioridad más alta;
 - si hay varios iguales: el primero de la cola
 - prioridades estáticas 1..99
- Dos clases de planificación en tiempo real para procesos con igual prioridad
 - FIFO por orden de llegada (SCHED_FIFO)
 - Se ejecuta hasta que termine (cuanto ilimitado)
 - Sólo es expropiable por procesos TR de más prioridad
 - Round Robin turno circular (SCHED_RR)
 - Es desalojado si agota su cuanto
 - Vuelve al final de la cola de listos (misma prioridad)
 - El tiempo se comparte en procesos TR con la misma prioridad



planificación Linux: Posix

Contenido

concepto criterios algoritmos ejemplos

Syscalls para planificación de procesos

- Modificar la prioridad de un proceso
 - int **sched_setparam**(pid_t pid, const struct_param *param);
- Modificar la prioridad y la política
 - int sched_setscheduler(pid_t pid, const sched param *param);
- Obtener los parámetros de planificación de un proceso
 - int sched getscheduler(pid t pid);
- Obtener la prioridad de un proceso
 - int sched_getparam(pid_t pid, const struct_param *param);



Windows NT

Contenido

concepto criterios algoritmos ejemplos

Política de planificación

- NT es un SO operativo expropiativo, donde el thread es la unidad básica de planificación.
- Planificación expropiativa en base a prioridades:
 - Estáticas para tiempo real [16..31], sólo el administrador
 - Dinámicas para usuario [1..15], aplicaciones típicas
- La prioridad 0 se asigna al thread nulo (ocioso)
- Procesos en tiempo real: tienen mayor prioridad
- Favorece threads interactivos incrementando su prioridad
- Tiempo compartido mediante cuantos de tiempo:
 - Determinación del cuanto de tiempo:
 - 2 ticks NT workstation 12 ticks NT server (20 120 ms), sobre x86.



Windows NT

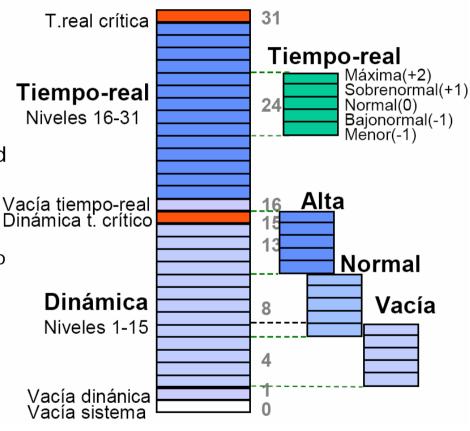
Contenido

concepto criterios algoritmos ejemplos

Prioridad base del proceso:

Realtime (24)

- High (13)
- Above Normal (10)
- Normal (8)
- Below Normal (6)
- Idle (4)
- Prioridad base del thread (offset max. ±2 de la priordad base del proceso):
 - Time_Critical (15 para todas las clases excepto RT que es 31)
 - Idle (1 para todas las clases excepto para RT que es 16)
 - Highest (+2)
 - Above normal (+1)
 - Normal (+0)
 - Below Normal (-1)
 - Lowest (-2)





Windows NT

Contenido

concepto criterios algoritmos ejemplos © Correspondencia prioridades Win32 - kernel

		Win32 process class priorities					
		Realtime	High	Above Normal	Normal	Below Normal	Idle
	Time critical	31	15	15	15	15	15
	Highest	26	15	12	10	8	6
Win32	Above normal	25	14	11	9	7	5
thread	Normal	24	13	10	8	6	4
priorities	Below normal	23	12	9	7	5	3
	Lowest	22	11	8	6	4	2
	Idle	16	1	1	1	1	1



Windows NT

Contenido

concepto criterios algoritmos eiemplos

- El planificador mantiene 32 colas para threads listos: una cola por cada prioridad
- Las prioridades dinámicas utilizan colas multinivel con realimentación
- Se escoge el primer thread listo de la cola de prioridad más alta no vacía
- Tiempo compartido en turno circular (RR) dentro de un mismo nivel de prioridad
- El thread expropiado
 - Por fin de cuanto se coloca al final de la cola de su prioridad (RR)
 - Por un thread de más prioridad se coloca en la cabeza de la cola de su prioridad (conserva el turno)



Windows NT

Contenido

concepto criterios algoritmos ejemplos

Ajustes automáticos de las prioridades

- Los threads de usuario (prioridades dinámicas < 16) pueden experimentar incrementos (boosts) y reducciones (decays) automáticos en su prioridad
- Para favorecer procesos intensivos en E/S y para evitar inanición
- Los threads de tiempo real (prioridad > 15) no experimentan estos ajustes
 - La planificación en TR es predecible: respeta la prioridad original de cada thread. Pero esto NO garantiza tiempos de respuesta.



Windows NT

Contenido

concepto criterios algoritmos ejemplos

- Incrementos en la prioridad de un thread
 - Tienen lugar tras un bloqueo, normalmente cuando se resuelve una petición de E/S
 - Dispositivos lentos = grandes incrementos (p.e. KB o ratón + 6)
 - Dispositivos rápidos = pequeños incrementos (p.e. HD + 1)
 - El incremento se aplica sobre su prioridad base
 - El resultado nunca es mayor que 15 (máx. para usuarios)
 - Buen tiempo de respuesta para interactivos, dispositivos de E/S se mantienen ocupados
- Decrementos en la prioridad
 - Cada vez que consuma un cuanto de ejecución, se decrementa en 1 la prioridad. Hasta alcanzar la prioridad base del thread



Windows NT

Contenido

concepto criterios algoritmos ejemplos

Eliminación de la inanición

- "Balance Set Manager" es un thread con prioridad 16 (de sistema) que se despierta cada segundo
- Busca threads de usuario que han estado listos 3 ó más segundos sin ejecutar
- Les incrementa la prioridad y el cuanto
 - Prioridad = 15 (máxima para usuario)
 - Doble cuanto de tiempo
- El incremento es transitorio: sólo hasta completar el cuanto o bloquearse, después vuelve a su prioridad y cuanto normal



Windows NT

Contenido

concepto criterios algoritmos ejemplos

- Incrementos automáticos de prioridad algo arbitrarios
 - Favorece los threads que usan dispositivos de E/S
 - Los incrementos están prefijados
 - La tarjeta de sonido tiene máximo incremento +8
 - Si queremos ir rápido hay que tocar mucha música
- El sistema no evita implícitamente la inanición
 - No existe envejecimiento, es necesario vigilar activamente que no ocurra postergación indefinida
- Soporte para tiempo real soft
 - No se garantizan tiempos de respuesta
 - Sólo que la planificación es predecible (prio. estáticas)



Windows NT

Contenido

concepto criterios algoritmos ejemplos Asignación de la prioridad base del proceso:

CreateProcess(...,fdwCreate,...)

- Aplicaciones a nivel de usuario
 - TASKLIST.EXE: listar procesos (como ps en unix)
 - TASKMGR.EXE: parecido, pero con interfaz gráfico
 - START /LOW file.exe: permite lanzar procesos
 - Otros parámetros son /Normal, /High o /RealTime
 - TASKKILL.EXE: termina el proceso especificado



Windows NT

Contenido

concepto criterios algoritmos ejemplos

Syscalls para planificación

- © Control de la clase de prioridad
 - BOOL **SetPriorityClass**(HANDLE hProcess, DWORD fdwPriorityClass);
 - DWORD GetPriorityClass(HANDLE hProcess);
- Control de la prioridad relativa del thread
 - BOOL SetTrheadPriority(HANDLE hThread, DWORD fdwPriority);
 - DWORD GetThreadPriority(HANDLE hProcess);

FSO - Examen septiembre de 2006

8.- En un sistema con planificación de colas de realimentación multinivel de tres niveles se ejecutan tres procesos.

Los tiempos de computación pura de los procesos (total CPU) son los siguientes: P1=60 ciclos, P2=130 ciclos y P3=180 ciclos. Además el proceso P1 realiza una operación de E/S cuando han pasado 20 ciclos de su tiempo de computación. El proceso P2 realiza una operación de E/S cuando han pasado 80 ciclos de su tiempo de computación. El proceso P3 no realiza operaciones de E/S. Cada operación de E/S dura 25 ciclos.

Los cuantos de tiempo asociados a cada cola, a medida que desciende la prioridad de las colas son Cola1: 50, Cola2: 100 y Cola3: 120 ciclos (la Cola1 es la de mayor prioridad). Suponiendo que, inicialmente, todos los procesos llegan al mismo tiempo al sistema de planificación y que el orden inicial en la cola es P1(primero), P2(segundo) y P3 (tercero), completar el diario de ejecución en la tabla siguiente anotando el tiempo en el cual ocurre cada nuevo evento, dónde se encuentra cada proceso en ese momento y la descripción del evento.

Tiempo	Ejecución	Cola1	Cola2	Cola3	Bloqueados	Descripción evento
0	P1	P2,P3		4	F .	Llegada de los procesos P1,P2, P3 Despacho de P1
		3				

Ejemplo de planificación

Proceso	Llegada	Tiempo CPU	Prioridad
T0	0	50	4 (min.)
T1	10	50	3
T2	20	20	1 (max.)
T3	30	10	2

 El proceso T0 se bloquea a las 10 unidades de tiempo durante 15 unidades de tiempo

	Expropiativa	
por orden de llegada	FCFS	Turno circular (RR) Q=10
por tiempo de ejecución	Trabajo más corto primero (SJF)	Tiempo restante más corto primero (SRTN)
por prioridades	Prioridades	Prioridades expropiativas

Ejemplo de planificación

Proceso	Llegada	Tiempo CPU	Prioridad
T0	0	50	4 (min.)
T1	10	50	3
T2	20	20	1 (max.)
T3	30	10	2

 El proceso T0 se bloquea a las 10 unidades de tiempo durante 15 unidades de tiempo

	Expropiativa	
por orden de llegada	FCFS	Turno circular (RR) Q=10
por tiempo de ejecución	Trabajo más corto primero (SJF)	Tiempo restante más corto primero (SRTN)
por prioridades	Prioridades	Prioridades expropiativas