Immagine che contiene testo, carta, menu, Prodotto di carta

Descrizione generata automaticamente

1. Le caratteristiche dello scheduler di CPU Unix tradizionale sono:
   1. La presenza di code multiple con feedback
   2. La gestione interna alle code in modalità Round Robin
   3. Un livello di priorità per ciascuna coda

In particolare, i livelli di priorità sono 40 e vanno in un range da -20 a +19.

Il livello più importante è il -20 ed è quello che il processore schedulerà prima di tutti, i restanti livelli vanno in ordine decrescente di priorità.

In generale, quando un processo entra nello stato Ready, gli viene attribuita una priorità base livello 0.

Il feedback, consiste appunto nella possibilità di spostare un processo da una coda all’altra in base a 2 eventi:

1. Il processo rientra nello stato Ready dopo essere stato nello stato Sleep
2. Periodicamente il sistema operativo può variare la sua priorità secondo la formula:

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>P</mi><mo>=</mo><mi>B</mi><mi>a</mi><mi>s</mi><mi>e</mi><mo>+</mo><mi>f</mi><mfenced><mrow><mi>C</mi><mi>P</mi><mi>U</mi><mo>&#xA0;</mo><mi>u</mi><mi>s</mi><mi>a</mi><mi>g</mi><mi>e</mi></mrow></mfenced><mo>+</mo><mi>n</mi><mi>i</mi><mi>c</mi><mi>e</mi></mstyle></math>"}

dove *f(CPU usage)* ci dice quanto il processo è CPU Bound, ossia quanto fa uso della CPU e

*nice* è un valore che rappresenta la niceness di un processo, ossia quanto è amichevole.

In questo modo, i processi maggiormente CPU Bound e amichevoli, si troveranno in cima alla scala delle priorità.

La nice varia da +19 a -20, se si prova a dare un valore superiore (inferiore) al margine, si finisce al

livello massimo (minimo).

Un processo con nice alto mandano in CPU gli altri processi.

Le system call offerte per la gestione di questo scheduler sono:

* **int nice(int incr)** questa è la system call che viene invocata dalla shell quando si passa sulla linea di comando: nice[number][command]
* **int getpriority(int which, int who)** ci dà la priorità di who in relazione di which che può riferirsi a uno tra processo, gruppo del processo o utente, se who=0 ci possiamo riferire al processo chiamante, al gruppo del processo chiamante o all’utente del processo chiamante
* **int setpriority(int which, int who, int prio)** possiamo settare la priorità con lo stesso meccanismo sopra e in più possiamo impostare la niceness con il parametro prio, di default vale 0

1. Nell’allocazione di file a catena, i blocchi di un file sono collegati come in una lista e non sono posizionati necessariamente in maniera contigua.

L’occupazione reale è sempre pari al numero di blocchi nella lista.

Il Record di Sistema mantiene informazioni solo sul primo blocco (nome del file – blocco di inizio), ogni blocco mantiene informazioni per arrivare al blocco successivo.

L’accesso al file è potenzialmente costoso se i blocchi sono molto lontani tra loro.

In questo caso la ricompattazione è utile per diminuire il costo di accesso.

Il vantaggio di questa tecnica è la non presenza di frammentazione esterna, poiché per ogni blocco libero posso allocare nuove informazioni del file.

File con 10k record allocato con lo schema a catena

Memoria di massa con blocchi di taglia 1k record con latenza per caricamento in memoria ram di 10ms

Calcolare latenza del caso peggiore per il caricamento in ram dell’ultimo record considerando che:

* il tempo richiesto dal software di sistema per l’identificazione di un indice di blocco di dispositivo (quando già caricato in memoria) sia pari a 1ms
* il record di sistema di ogni file sia contenuto in un unico blocco del dispositivo
* l’indicizzazione dei blocchi di dispositivo richieda l’utilizzo di indici di taglia pari a 64 record

Immagine che contiene testo, calligrafia, Carattere, numero

Descrizione generata automaticamente

1. Ad ogni frame è associato un reference bit, ossia un bit che viene impostato a 1 quando la pagina in quel frame viene referenziata in lettura o scrittura.

Periodicamente, il sistema operativo scorre, con una lancetta del reset, l’insieme dei reference bit e li imposta a 0.

Quando è necessaria una sostituzione di pagina, il sistema operativo scorre i reference bit con la lancetta di selezione finché non ne trova uno impostato a 0.

A quel punto, la pagina corrispondente viene selezionata per la sostituzione e tutti gli altri reference bit a 1 vengono resettati.

L’algoritmo dell’orologio soffre dell’anomalia di Belady perché non è un algoritmo a stack, in particolare cerca di approssimare il risultato all’algoritmo LRU che è invece un algoritmo a stack.