

La tecnologia di Bilanciamento dei Processi di Process Lasso's (ProBalance)

Indice

1. [Com'è](#) condivisa la CPU tra tutti i processi in esecuzione?
2. Qual è la tecnologia di [Process Lasso 'ProBalance'](#)?
3. Perché non funziona aumentare la priorità del processo in primo piano?
4. [E' utile ProBalance nei PC multicore?](#)
5. [Devo priorizzare tutti i miei processi, classificandoli secondo l'importanza per me?](#)
6. [Perché Microsoft non ha progettato lo Scheduler della CPU di Windows per gestire meglio gli scenari di carico estremo della CPU? \[per gli scettici\]](#)
7. [Dimostrazione grafica](#)
8. [Link di terze parti](#)
9. Domande [Frequenti](#) (in un'altra pagina)

Com'è condivisa la CPU tra tutti i processi in esecuzione?

Un processo ha un certo numero di thread (essenzialmente task in esecuzione). Questo è ciò che vede ed esegue il sistema operativo. Fondamentalmente tiene una lunga lista dei thread e passa da uno all'altro, dando ad ognuno un turno per usare la/le CPU per un breve periodo di tempo (una time slice). Qui è dove agisce la priorità dei thread, insieme ad altri fattori. La priorità aiuta lo scheduler (la parte del SO responsabile della distribuzione delle time slice) determina quale thread deve prendere la prossima time slice.

La CPU può fare solo una cosa per volta. Quindi, per fare multitask, la CPU è commutate velocemente tra tutti i thread attivi nel sistema. E' in grado di commutare tra i thread così velocemente che sembra che i processi siano in esecuzione contemporaneamente, quando in realtà vengono eseguiti a turno. Nel caso di sistemi multi-CPU (multi core), ciò è sempre vero, eccetto che ci sono più CPU disponibili per l'esecuzione dei thread, permettendo qualche vera esecuzione simultanea e non a turni.

Quando i thread di processi multipli hanno bisogno di usare la CPU allo stesso tempo, il livello di priorità di ciascun thread viene usato per decidere quale deve essere il seguente ad usare la CPU (qual è il più importante). Abbassando temporaneamente la classe di priorità dei processi in secondo piano eccessivamente attivi, il vostro sistema è mantenuto reattivo anche in caso di alti carichi della CPU. Abbassando la classe di priorità di un processo, si abbassano le priorità di tutti i thread di quel processo. Quando si agisce su di un processo, viene cambiata la priorità di base da cui vengono derivate le priorità dei singoli thread. In questo modo, le priorità di ogni thread in un processo sono le stesse.

Qual è il problema?

Windows ha particolari difficoltà ad interagire con thread che decidono di consumare ogni microsecondo del tempo CPU su cui riescono a mettere le mani (un thread CPU bound). Un solo thread "CPU bound" eseguito a priorità Normale può portare un sistema con una singola CPU al blocco totale, come si può vedere nella nostra dimostrazione grafica più sotto. E' proprio così, ci crediate o no! Process Lasso fu originariamente scritto per risolvere questo scenario. Abbassando temporaneamente la priorità del processo "mangione", il vostro PC può essere salvato da uno stallo virtuale. E' stato poi scoperto che con alcuni perfezionamenti il nostro algoritmo poteva migliorare la reattività del sistema durante periodi di alto carico della CPU – oltre a salvare il PC da uno

scenario ancora peggiore.

Alcuni scheduler, come quello predefinito del kernel Linux, penalizzano i thread “CPU bound” mentre favoriscono i thread “I/O bound”. Ciò è simile a quello che fa ProBalance. Nei sistemi Windows **la maggior parte** dei thread sono “I/O bound”. Essi abbandonano la loro time slice prematuramente perchè entrano in uno stato di attesa dovuto a qualche tipo di I/O. Sfortunatamente i thread “CPU bound” di tanto in tanto entrano in funzione e “si fanno polpette” dei meno voraci thread “I/O bound”.

Questo non vuol dire che dovete cominciare a ridare una nuova priorità ai processi in esecuzione, ordinandoli secondo le vostre priorità. Lasciate che il nostro algoritmo ProBalance lo faccia automaticamente. NON fate regolazioni manuali delle priorità a meno che sappiate davvero cosa state facendo. Ciò può creare più danni che vantaggi. Noi abbiamo scritto il nostro software per operare in sicurezza ed efficacemente nella sua configurazione predefinita ma non possiamo dire lo stesso per tweak personalizzati dall’utente.

Perché non funziona aumentare la priorità del processo in primo piano?

Primo, per quelli che non lo sapessero, il processo in primo piano è quello che possiede la finestra che ha il focus del mouse e della tastiera. Ciò vuol dire che c’è sempre un solo processo in primo piano per volta. E’ di solito il programma con cui state lavorando. Alcuni programmi di qualità inferiore credono che aumentare la priorità del processo attivo in primo piano sia utile ad aumentare le prestazioni e la reattività. La verità è che non fanno nulla di tutto ciò ma spesso fanno il contrario.

Windows applica già un meccanismo che favorisce i programmi in primo piano, dando al thread in primo piano una “time slice” più lunga. Aumentare ulteriormente la priorità del processo in primo piano e/o uno specifico thread in primo piano non è efficace per nulla. Ricordate, dare ad un processo una priorità più alta non significa che verrà eseguito più velocemente. Significa semplicemente che se diversi processi sono attivi nello stesso tempo, lui avrà la precedenza sugli altri. Comunque, se un singolo processo ha una precedenza troppo alta sugli altri processi, siccome si aggiunge anche la precedenza come processo in primo piano, ci possono essere delle complicazioni. Infatti la nostra ricerca ha evidenziato che un’eccessiva precedenza del processo in primo piano può creare un certo numero di complicazioni con i component del SO e varie applicazioni.

Siate molto cauti se un utility dice di aumentare le prestazioni di un PC, aumentando la priorità del processo in primo piano. Queste utility sono estremamente facili da creare e sono una delle tante dichiarazioni ad effetto che alcune persone tentano di spingere. In alcuni casi l’autore del programma non capisce semplicemente il funzionamento dello scheduler della CPU e perché ciò sia una cattiva idea. In altri casi le compagnie “se ne fregano” e tentano solo di fare soldi. In breve questa è una soluzione completamente sbagliata che deve essere evitata a tutti i costi.

Qual’è la tecnologia di Process Lasso “ProBalance”?

ProBalance, che sta per Bilanciamento dei Processi, è il nome del nostro algoritmo proprietario che temporaneamente regola la priorità dei processi in esecuzione nello sforzo di mantenere il vostro sistema reattivo durante alti carichi.

Quando c’è un carico alto sulla/e CPU del sistema, la reattività del PC può essere severamente compromessa. In un PC con un processore singolo, basta un solo thread “ingordo” funzionante a priorità normale per portare l’intero sistema allo stallo. Anche thread ad alta priorità possono essere lasciati “all’asciutto” da un thread “ingordo” che funziona a priorità normale. Questo problema è inerente a come è progettato lo scheduler della CPU di Windows e dall’architettura generale del

SO. Parrebbe che un tale problema non dovesse esistere, ma purtroppo c'è e può essere facilmente dimostrato (leggere la sezione “scettici” più in basso).

L'utente comune si accorge spesso di una minore reattività del suo PC durante brevi periodi di tempo, chiamati micro lag. Qualche volta ciò succede per un errore di un processo in secondo piano ma altre volte succede solo perché c'è un alto carico sulla CPU. Sicuramente avete sperimentato questa condizione – il movimento del mouse diventa a singhiozzo e tutte le azioni sono estremamente lente.

ProBalance di Process Lasso abbassa con intelligenza la classe di priorità dei processi in secondo piano che potrebbero interferire con la reattività del sistema. Ciò non limita le prestazioni dei processi in secondo piano perché comunque ottengono una quantità considerevole del tempo CPU disponibile, ma aumenta tremendamente la reattività del processo in primo piano. Dopo tutto ci vogliono pochi cicli CPU per mantenere il processo in primo piano reattivo. Prenderli dai processi in secondo piano, quando necessario, non crea a loro nessun problema. Anche perché questo accorgimento è temporaneo ed è annullato appena le condizioni del sistema cambiano. Process Lasso è progettato per essere intrusivo al minimo, abbassa le priorità solo quando è appropriato e si assicura che i processi in secondo piano continuino a funzionare bene.

E' utile ProBalance nei PC multicore?

Certo. Dato l'ampio uso che facciamo del multi-tasking, ProBalance continua ad aiutare pesantemente la reattività anche nelle CPU multi-core. Per esempio, mentre scrivevo Process Lasso, e avevo una macchina virtuale in funzione (come test), la differenza con e senza Process Lasso è stupefacente in una CPU x86. I PC multicore sono soggetti allo stesso tipo di problemi delle CPU a core singolo, dato che devono eseguire un numero maggiore di thread “mangioni” che inficiano grandemente la reattività del sistema. Naturalmente Process Lasso offre anche una quantità di caratteristiche oltre a ProBalance che sono utili ai possessori di PC multicore.

I possessori di multi-core trovano molto utili alcune funzionalità, come l'affinità della CPU predefinita, il cambio del watchdog dell'affinità della CPU e il cambio dell'affinità della cpu di ProBalance.

Devo prioritizzare tutti i miei processi, classificandoli secondo l'importanza per me?

Assolutamente **NO**. Qualche programma vi incoraggia a fare questo, ma è pericoloso. Gli autori di questi programmi non sono probabilmente al corrente dei pericoli, oppure non gli importa. Anche se Process Lasso permette di bloccare le priorità predefinite, questa funzione deve essere usata solo in certe circostanze appropriate. Se non siete un esperto, è meglio che l'algoritmo attentamente progettato di ProBalance sistemi dinamicamente la priorità dei processi quando necessario.

Dimostrazione della teoria

Con tutti I programmi spazzatura disponibili su internet, come i deframmentatori della RAM, è natural per alcuni utenti essere scettici. Le critiche più comuni vengono da quelli che hanno un'eccessiva fiducia nelle prestazioni dello scheduler della CPU di Windows. Queste persone credono che se lo scheduler di Windows fosse scritto come dovrebbe essere, di sicuro i thread a priorità normale non possono interferire con altri thread con priorità normale (e alta). Questo argomento ha risonanza particolarmente in gruppi non-Windows che non hanno esperienze delle prestazioni reali dello scheduler della CPU di Windows.

La maggior parte degli sviluppatori in Windows vi diranno che sono delusi dallo scheduler della

CPU di Windows. Vi racconteranno di quella volta in cui avevano un bug che causava un loop infinito in un thread con priorità normale eseguito in un processo con classe di priorità normale. Con sgomento vedevano il loro sistema completamente in stallo. Non potevano crederci ma lo hanno visto coi loro occhi.

La verità è che, in un sistema con una singola CPU, un solo thread con priorità normale può portare l'intero PC quasi al congelamento di tutte le sue funzioni! Perfino thread con priorità alta possono avere "fame" di cicli CPU a causa di un thread "affamato" a priorità normale. Proprio così, stupefacente ma vero. Nei sistemi multi-CPU, il numero dei thread necessari per portare il sistema ad uno stallo complete è uguale al numero delle cpu disponibili. Non ci credete? Provate la nostra applicazione dimostrativa o scrivetene una voi. Ogni thread che non fa altro che "mangiarsi" cicli CPU senza ritegno, dimostrerà questo fenomeno.

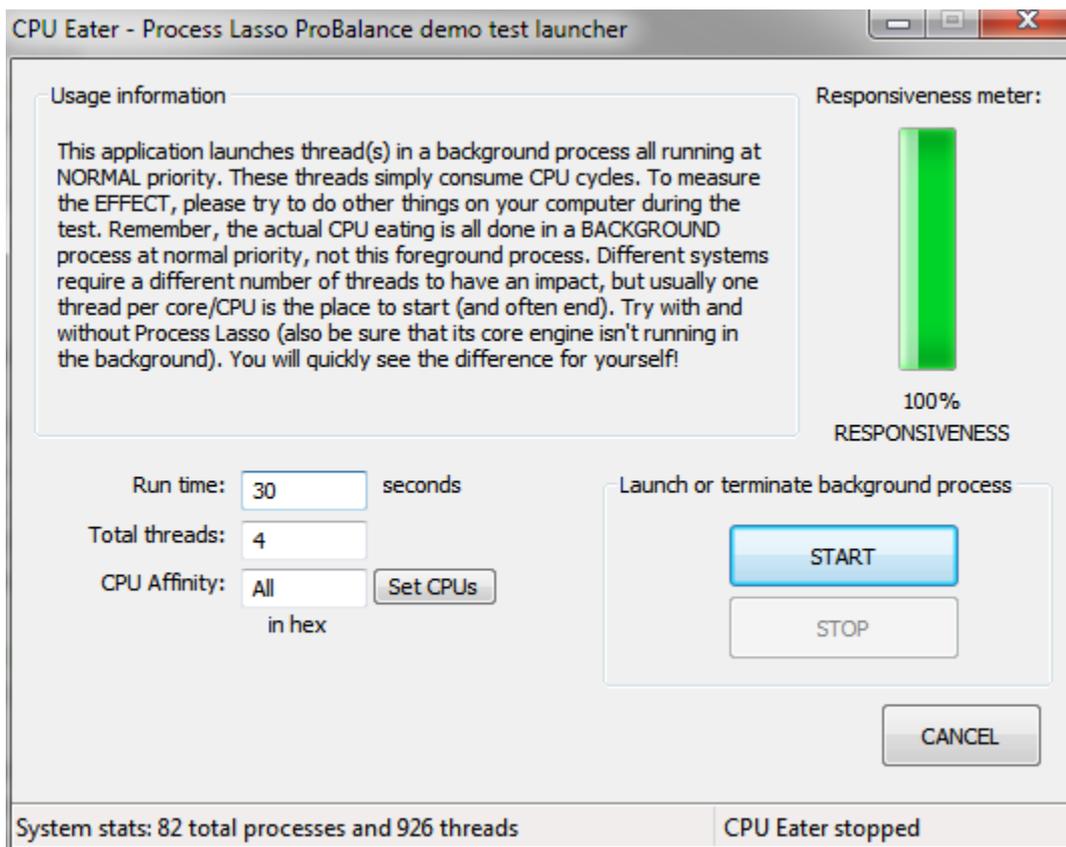
Dimostrazione pratica

Abbiamo sviluppato un programma "mangiatore di CPU" con cui potete provare Process Lasso. Questo "CPU Eater" crea una manciata di thread, tutti funzionanti a priorità NORMALE. Questi thread usano solo i cicli della CPU. Quindi, *se* lo scheduler della CPU di Windows si comportasse come vi aspettereste, gli altri processi in esecuzione avrebbero abbastanza cicli della CPU per rimanere reattivi – anche quando questo "CPU Eater" funzionasse in secondo piano. Sfortunatamente questo non è il risultato. La reattività del vostro computer sarà sostanzialmente degradata mentre "CPU Eater" è in funzione! Controllate coi vostri occhi...

Questo non è un trucco! Potete facilmente creare il vostro "CPU eater", o provare utilità simili, invece del nostro. Qualche CPU Eater disponibile nella nuvola incrementano la loro priorità, perché non si rendono conto che basta una priorità Normale per "fare il lavoro". Per questi programmi, siate sicuri di disabilitare l'opzione di ProBalance "Opzioni / Impostazioni ProBalance / Escludi i processi che non hanno priorità normale" o Process Lasso li ignorerebbe. Normalmente, Process Lasso lascia stare i processi a priorità alta perché pensa che abbiano aumentato la loro priorità per una buona ragione. Ancora una volta, noi tentiamo di agire con INTELLIGENZA, che è la chiave per fare del buon software. Quindi createne uno VOSTRO. Un loop infinito scritto in qualsiasi linguaggio è perfetto.

Questa demo "ha la mano pesante" coi valori predefiniti ma potete abbassare il numero dei thread senza problema. Basta molto meno dei valori predefiniti! RICORDATE che i thread creati sono in secondo piano. Thread "fuori controllo" in primo piano creerebbero "maggiori danni" dal momento che avrebbero time slice più lunghi (dipendenti dalla configurazione dello scheduler e dal tipo di SO).

- [Download 32-bit CPU Eater](#)
- [Download 64-bit CPU Eater](#) (solo per Windows x64)



Istruzioni d'uso:

1. Download ed eseguite dal link più in alto appropriato per il vostro sistema operativo (32 o 64 bit)
2. L'applicazione si aprirà automaticamente. Chi ha Vista o Windows 7 vedrà aprirsi una finestra di autorizzazione del sistema UAC. Confermare e si aprirà una finestra come quella sopra
3. Cliccate il bottone START per iniziare il processo "mangia CPU" in secondo piano e con priorità NORMALE
4. Se il vostro PC sembra congelato perché ProBalance di Process Lasso non è attivo, ASPETTATE e tutto tornerà normale quando il test terminerà. Abbiamo anche aumentato la priorità dell'applicazione di controllo in modo da riuscire a premere il tasto STOP in qualsiasi momento del test per terminare il processo che in secondo piano impegna tutti i cicli della CPU. Naturalmente se Process Lasso è in esecuzione e ProBalance è abilitato, non avrete nessun problema.

Ricordatevi che se non avete installato Process Lasso, il test mostrerà semplicemente le mancanze dello scheduler della CPU di Windows. Dovete installare Process Lasso per vedere I benefici di ProBalance.

Dopo aver installato Process Lasso, disabilitate ProBalance (o chiudete ProcessLasso E il suo motore ProcessGovernor). Quindi avviate CPU Eater per vedere come si comporta il vostro computer con ProBalance disabilitato. Portato a termine questo primo test, riabilitate ProBalance o riavviate Process Lasso. Ora eseguite di nuovo lo stesso test. Sarete impressionati dalle differenze senza distinzioni di quanti core abbia la CPU del vostro computer!

Ricordate che i thread del processo in secondo piano lanciati da CPU Eater funzionano tutti con priorità NORMALE, e non fanno nient'altro che eseguire un loop senza fine. Non viene fatta nessuna ulteriore regolazione, lasciamo che Windows le gestisca. Tutte le applicazioni in esecuzione sul vostro computer hanno il potenziale di generare gli stessi effetti che procura CPU

Eater.

ATTNEZIONE: come precauzione vi raccomandiamo di salvare tutti I documenti aperti, prima di eseguire questo test. Inoltre non resettate il vostro computer quando state provando CPU eater senza ProBalance. Il vostro computer comincerà a rispondere meglio dopo 60 secondi. L'uso di questo programma di test è a vostro rischio e pericolo, anche se non ci sono rischi particolari ma il vostro sistema verrà sottoposto ad un carico pesante che potrebbe causare problemi a sistemi che non abbiano un adeguata dissipazione del calore. Siamo obbligati ad avvertirvi, quindi usate questo test sapendo cosa state facendo.

Il semplice test più sopra ci mette 2 minuti. Provatelo!

Come funziona il nostro test

E' facile dimostrare le mancanze dello scheduler di Windows. Dapprima, per questioni di semplicità, inizieremo a descrivervi un sistema che ha una singola CPU. La dimostrazione può essere scalata facilmente ai core multipli ma è più facile descrivere il concetto con una sola CPU.

Cominciate creando un processo che non fa nient'altro che eseguire un loop infinito, così da consumare tutti I cicli della CPU disponibili per lui. Lo chiameremo "CPU eater". Quindi lanciate questo processo come qualsiasi altra applicazione, con una classe di priorità normale. Se lo scheduler della CPU di Windows funzionasse come uno si aspetta che facesse, la prestazione degli altri processi nel sistema, e il processo in primo piano, non dovrebbero diminuire. Sfortunatamente noterete che questo singolo processo che funziona a priorità normale è in grado di togliere cicli CPU agli altri processi e portare il sistema vicino al blocco totale – perfino terminare questo processo è molto difficile. Ciò è vero anche con lo scheduler della cpu di Vista e Windows 7, nonostante sia stato migliorato.

Per scalare questo argomento in caso di CPU multiple, semplicemente programmate il processo di "CPU eater" per avere thread multipli o lanciate diverse istanze del processo di CPU eater. Vedrete lo stesso effetto di prima.

Gli aggiustamenti manuali possono aiutare

Usando alter grandi funzioni di Process Lassos, come le priorità ed affinità predefinite dei processi, potete veramente mettere a punto le prestazioni del vostro sistema. ProBalance offre la grande possibilità di aumentare la reattività del vostro sistema automaticamente, ma se desiderate ulteriori miglioramenti, Process Lasso ve lo permetterà. Nessun algoritmo automatizzato può rimpiazzare l'intelligenza umana.

Perché Microsoft non ha sistemato questo problema?

Una ragione è che il comportamento previsto di un'applicazione Windows non è quello che si osserva nel mondo reale. Il loro intendimento sarebbe quello di fare un breve picco di uso della CPU e poi entrare in uno stato di attesa. Invece i programmi sono spesso mal progettati e portano ad una situazione di CPU al 100% e PC quasi bloccato. E' impossibile per Microsoft conoscere con sicurezza quale applicazione ha veramente bisogno di quei cicli CPU e quale no. Con un po' di "buon senso", un po' di logica e qualche aiuto dagli utenti, ProBalance di Process Lasso può fare meglio – salvando il vostro PC da uno stallo totale o semplicemente aumentando la reattività.

Cosa rimane dopo la disinstallazione?

Chiedete pure a quelli che hanno provato Process Lasso, ma poi lo hanno disinstallato. Cosa è rimasto? **Niente**. Nessun cambiamento permanente del sistema è stato fatto e non rimane nessun

file. Come tutti voi, anche a noi non piace lasciare “spazzatura” in giro dopo una disinstallazione e specialmente non ci piacciono i software che fanno cambiamenti permanenti alla configurazione del sistema e che non sono ripristinati dopo la disinstallazione. Come nota a margine, Process Lasso NON fa nessun aggiustamento della vostra configurazione di sistema, gestisce semplicemente i processi SOLO quando il motore interno è in funzione. I tool avanzati inclusi in Process Lasso, TweaksScheduler e il Multimedia Scheduler Configuration Utility FANNO cambiamenti permanenti ma vengono creati prima dei backup per assicurare che possiate tornare alla situazione precedente.

UN FORTE AVVERTIMENTO E EULA

Process Lasso fu creato con una regola ben precisa in mente: ”Non arrecare Danni”. Quindi il suo algoritmo ProBalance opera solo cambiamenti ritenuti completamente sicuri. Inoltre, quando un cambiamento è necessario, viene fatto il minimo possibile. Descriverei questo come un approccio conservativo. Abbiamo visto altri software non essere così “gentili”, che ci crediate o no. State attenti ai programmi che come espediente aumentano le priorità con noncuranza. Molti di questi aumentano la priorità del processo in primo piano ad Alto che non solo non aiuta, ma è anche insicuro. Provate queste utility con la nostra demo – o scrivete la vostra “CPU Eater” se non vi fidate della nostra, non potrebbe essere più facile crearla. In C, un 'while(1);' (infinite loop) è tutto quello che ci vuole...

Quando farete cambiamenti manuali a certi processi di sistema, o abiliterete certe funzioni di Process Lasso, sono mostrate delle avvertenze che vi informano dei possibili rischi. Comunque non possiamo prenderci carico di ogni possibile errore indotto dall’utente o configurazione del software, quindi noi vi consigliamo di NON cambiare le impostazioni predefinite a meno che siate degli esperti. Cambiare con noncuranza le classi di priorità dei processi può avere potenzialmente effetti negativi sulla stabilità del vostro PC. Vi raccomandiamo di lasciar fare il proprio lavoro all’algoritmo ProBalance di Process Lasso, a meno che non sappiate bene cosa state facendo.

Dal momento che Process Lasso permette queste modifiche avanzate della configurazione, dobbiamo mostrarvi una diffida legale durante l’installazione (una EULA). Come sempre, vi assumete tutti i rischi per ogni possibile danno, tangibile o intangibile, risultante dall’uso giusto o sbagliato del nostro software. Se non foste d’accordo, allora noi non possiamo permettervi l’uso del nostro software. L’EULA completa è mostrata prima dell’installazione; se leggendola non siete d’accordo, dovete cancellare l’installazione.

Ricerche e link (esterni e interni in inglese)

Le questioni che ProBalance di Process Lasso risolve sono ben documentate in tutto internet. Siccome potremmo essere giudicati come di parte (anche se siamo di parte solo verso la verità), abbiamo deciso di pubblicare alcune di queste ricerche fatte da terze parti. Qui segue qualche link veloce e presto seguiranno altri.

- [Everything you never wanted to know about OS multi-tasking \(and why out of control processes need priority adjustment\)](#)

"I should also note that Windows has other scheduling complications, such as "foreground" and "background" tasks [...] However, the bottom line, in my humble opinion, is that the operating system is more concerned in ensuring that a task that is not ready to run does not, than making sure all tasks get a fair shake. [...] We can, however, detect thread and process CPU utilization, and help the operating system to adjust the prioritizations."

-(c) Tim Murgent of TMurgent Technologies

- [Hardware Virtualization: the Nuts and Bolts](#)

"One process that takes up 100% of the CPU time may slow the other applications to snail speed for example, despite the fact that modern OSES use preemptive multitasking."

-(c) Johan De Gelas, AnandTech

- From a former competitor that popped up briefly:

"Priorities and their purpose are often poorly understood. Usually the misapprehension exists that if a process is taking a long time to complete then, if anything, its priority should be increased with the intent to make it finish faster. In most cases the CPU is spending near to all of its time executing the process anyway, so increasing the priority is only going to tell the scheduler to do what it is already doing. It may seem paradox but it is in fact more sensible to reduce the priority of the busy process so that the system is prepared to let other processes have some time thus allowing them to remain functional. In most cases other processes will not be using a significant amount of the CPU's time and allowing them to execute when they want to will not slow the busy process by any notable amount. Computer applications simply have to be idle practically all of the time in order for them to react quickly when the user interacts with them. Therefore, allowing a well behaved application to continue to respond in such a way will not take a significant amount of time away from the busy process."

-(c) Peter Ball

- Quote from [Inside the Linux scheduler \(IBM\)](#) - an example of another OS scheduler's attempt to address this

To prevent tasks from hogging the CPU and thus starving other tasks that need CPU access, the Linux 2.6 scheduler can dynamically alter a task's priority. It does so by penalizing tasks that are bound to a CPU and rewarding tasks that are I/O bound. I/O-bound tasks commonly use the CPU to set up an I/O and then sleep awaiting the completion of the I/O. This type of behavior gives other tasks access to the CPU.

Articoli correlati

- [The Linux Scheduler](#) (a similar scheduler to Windows, but tweaked differently)