

Corso Di Linux

Liceo Fermi, Bologna - Marcello Galli, novembre 2009

Funzioni dei sistemi operativi

Il sistema operativo (la cui parte centrale e' detta kernel) e' quel programma che gestisce il computer. Le sue principali funzioni sono essenzialmente:

- gestione dei processi
- gestione della memoria
- gestione accesso alle periferiche (tramite i drivers)

Gestione dei Processi

In un computer sono contemporaneamente in esecuzione parecchi programmi, questi programmi, con tutti i loro dati e le loro tabelle sono chiamati in gergo tecnico "processi". I processi sono tanti, la CPU pero' e' una sola, ed anche se in certi computer moderni ce ne sono 2 o perfino 4, sono sempre meno dei processi che corrono contemporaneamente in un computer. La soluzione e' mettere i programmi in coda, alcuni accedono alle CPU e corrono, altri aspettano il loro turno. La coda deve essere gestita in base a criteri di priorita' ben studiati, in modo che ogni processo possa avere usare la CPU quando gli serve, senza far aspettare troppo gli altri. La gestione delle code si chiama "scheduling" ed e' uno dei lavori principali del sistema operativo. C'e' poi un modo per interrompere il programma che usa la CPU e dare il turno ad un altro programma, il meccanismo degli "interrupt", gestito direttamente dall'hardware. I processi a volte hanno anche bisogno di scambiarsi informazioni e di sincronizzarsi, anche queste interazioni fra processi sono gestite dal kernel

Gestione della Memoria

Nel computer c'e' un sistema di memoria "virtuale": ad ogni processo viene fatto credere di avere a disposizione circa 4Gb di memoria. Ma non e' vero, la memoria vera e' puo' essere molto meno, il trucco funziona perche' la memoria viene divisa in sezioni di alcuni kilobyte (dette pagine) ed assegnata ai processi solo quando i processi ne hanno veramente bisogno; difficilmente i processi hanno bisogno di tutti i 4Gbyte che credono di avere a disposizione. C'e' tutto un complicato sistema di tabelle (page table) per assegnare a turno memoria vera ai processi solo quando ne hanno bisogno e ricordare a che processo appartiene ogni pagina. Quando non c'e' abbastanza memoria vera, si rimedia utilizzando, come fosse memoria, una particolare area del disco rigido, detta "swap". Vi accorgete subito se state usando lo swap perche' il computer diventa lento, pero' tutto va avanti lo stesso. I dati ed i programmi viaggiano quindi di continuo fra disco, memoria vera, area di swap; per questo non si puo' spegnere semplicemente il computer usando l'interruttore, se lo si fa tutti i dati ed i programmi in viaggio fra memoria, disco e swap si perdono ed il disco viene scarabocchiato. Tutto questa gestione di pagine di memoria e' fatta dal sistema operativo.

Accesso alle Periferiche

Solo il sistema operativo accede alle periferiche, come tastiera, mouse, video, stampanti etc. I programmi no. Altrimenti un programma con errori potrebbe bloccare tutto il computer. Era quello che succedeva con le vecchie versioni di Windows; con Linux solo il programma sbagliato si blocca, ma tutto il resto continua ad andare.

Quando un programma ha bisogno di accedere ad una periferica chiede al kernel di farlo per lui.

Facciamo un esempio: un programma che ha bisogno di scrivere qualcosa sullo schermo; le cose vanno circa così:

- il programma si interrompe e manda un messaggio al kernel (un interrupt) dicendogli dove ha messo le cose da scrivere.
- Il kernel interviene, ed utilizza un programma apposito (il driver) per accedere alla periferica e scrivere sullo schermo, poi, siccome la periferica è lenta rispetto alla CPU, il kernel si interrompe da solo e dà il turno ad un altro processo.
- Quando la scritta è apparsa sul video la scheda video manda un segnale di interruzione alla CPU, di nuovo interviene il kernel, e torna a dare il turno al programma iniziale.

Tutte le volte che pigiamo un tasto o clicchiamo o muoviamo il mouse succede una cosa di questo tipo. L'hardware che pilota il mouse o la tastiera manda un segnale di interruzione alla CPU, interviene il kernel, che manda il segnale del tasto al programma giusto e poi gli dà accesso alla CPU, secondo le priorità previste dagli algoritmi di scheduling.

Diversi sistemi operativi

Ci sono diversi tipi di sistemi operativi, organizzati in modo diverso. Il problema è che un sistema operativo è molto complicato e le interazioni fra le sue diverse parti vanno progettate bene, altrimenti le prestazioni del sistema calano. Negli anni 80 si pensava che un sistema operativo ben fatto dovesse essere composto da una parte centrale piccola (micro-kernel) e molte parti ausiliarie, staccate.

Linux non è fatto così, è un pezzo unico (kernel monolitico). Linus diceva che i micro-kernel non sono efficienti nel comunicare con le parti staccate del sistema operativo e sono una bella idea teorica, ma non funzionano in pratica. È famosa una lite, fatta via internet, fra Linus e Tannenbaum (l'autore di Minix), dove Tannenbaum parlava malissimo di Linux perché non era una architettura a micro-kernel. Resta il fatto che a tutt'oggi vere architetture micro-kernel non se ne trovano in giro, e tanti sistemi operativi partono con l'idea del micro-kernel, e poi non funzionano bene e scendono a compromessi (come Windows-XP).

Linux è quindi un pezzo unico (kernel monolitico), ma composto da parti smontabili (moduli). La maggior parte dei driver sono infatti realizzati come parti smontabili; li inserite nel vostro kernel solo se servono, mentre il computer va. Quando non servono li potete togliere, senza riavviare il computer. Il sistema di inserimento moduli è spesso automatizzato: il kernel cerca lui quali moduli servono, li inserisce quando servono e li toglie quando non servono più. Se ad esempio inserite una chiavetta usb il kernel la vede, e carica il driver giusto per farla usare. Quando smontate la chiavetta il kernel toglie il driver inutilizzato.

© Marcello Galli, Novembre 2009. Sito di riferimento: <http://www.helldragon.eu>