**Che cos’è una syscall e come viene implementata?**

Le chiamate al sistema operativo costituisco un’interfaccia tra il processo e il sistema operativo.

Tali chiamate sono generalmente disponibili sotto forma di routine scritte in C o C++, sebbene per alcuni compiti di basso livello, come quelli che comportano un accesso diretto all’hardware, sarebbe necessario il ricorso ad istruzioni in linguaggio assembly.

Spesso ci sono degli involucri intorno a queste system call costituite da librerie opportune che permettono di operare in maniera agevole quindi esiste un’interfaccia per il programmatore (API) che in qualche maniera permette di richiedere servizi direttamente o indirettamente al SO.

Le funzioni fornite dalle API invocano solitamente le chiamate di sistema per conto del programmatore.

Le system call vengono implementate eseguendo delle trap o software interrupt, sono dell’istruzioni apposite che permettono di attivare il meccanismo dell’interruzioni.

Quindi le chiamate al sistema sono utilizzate dai programmi di sistema e dai programmi d’utente per interagire con il sistema operativo.

**Differenza tra un sistema operativo a kernel monolitico e microkernel?**

Il kernel fornisce il file system, CPU scheduling la gestione della memoria e altre funzioni del sistema operativo.

Quello che si trova al di sopra del kernel comunica con esso attraverso le system call e sopra il kernel ci sono shell, compilatori, interpreti e librerie di sistema e sopra ancora ci sono gli utenti.

Il metodo tradizionale di implementare un sistema operativo consiste nel mettere sopra la HW uno strato di software detto kernel che in qualche maniera fornisce l’interfaccia per le system call e fornisce diversi servizi i quali sono gestione del terminale, sistema di I/O, file system, CPU scheduling, rimpiazzo delle pagine, memoria virtuale; questo software è sempre caricato in memoria e viene attivato sulle interruzioni, quando arrivano interruzioni da parte dell’HW oppure quando qualcuno si interfaccia con il kernel mediante delle trap (si tratta di interrupt generate via software che in pratica sono il metodo che viene utilizzato per implementare le syscall).

Il kernel si mette in funzione quando arrivano le interruzioni che sono o quelle provenienti dall’HW o quelle genrate dalla syscall che servono per implementare le syscall.

Nel momento in cui eseguo una system call attivo il SO che passa in modalità protetta.

Il metodo tradizionale di implementare il sistema il SO ogniqualvolta che io richiedo un servizio al SO tutto quello che richiedo attraversa l’interfaccia delle system call e viene eseguita in modalità protetta (tutto quello che si trova al di sotto dell’interfaccia delle syscall viene eseguito in modalità protetta) dopodiché si ritorna dalla syscall e si ritorna dai programmi utente.

Altro metodo utilizzato per implementare il sistema operativo consiste nel inserire sopra l’HW uno strato di software di piccole dimensioni che si chiama microkernel il quale fornisce delle funzionalità minime, il minimo indispensabile che mi permette di far eseguire i programmi e farli comunicare tra di loro.

Le funzionalità che si svolgevano all’interno del kernel sono realizzate mediante dei programmi di sistema i quali girano non in modalità supervisore ma in modalità utente.

Il microkernel fornisce una frazione dei servizi che fornisce il kernel, tutto il resto del sistema operativo è implementato mediante dei programmi di sistema che girano in modalità utente.

I vantaggi che offre il microkernel è la facilità di estensione i nuovi programmi si aggiungono allo spazio utenti e non comportano modifiche al kernel inoltre poiché i servizi si eseguono in modalità utente ci permette una maggiore protezione, se un servizio è compromesso il resto del sistema operativo rimane intatto.

Svantaggi pago in prestazioni, poco performante perché richiede delle comunicazioni in più, quello che accadeva nel kernel facendo delle chiamate ad una funzione qui avviene con uno scambio di messaggi.

Il microkernel deve avere un minimo di funzionalità di gestione della memoria, deve avere i driver di I/O, lo scheduling della CPU perché è quella parte del SO che mi permette di lanciare più programmi in esecuzione e un minino di comunicazione dei processi.

Si descrivano le architetture dei kernel monolitici e di quelli a microkernel. Risposta: Le architetture dei kernel monolitici inglobano in questi ultimi praticamente tutto ci`o che sta fra l’hardware e le chiamate di sistema. Se da un lato ci`o permette di ottenere alte prestazioni, dall’altro rende il kernel poco flessibile in quanto non si distingue tra meccanismi (come fare qualcosa) e politiche (cosa deve essere fatto). Al massimo vi `e una stratificazione in livelli in cui ognuno di questi ultimi fornisce delle primitive di base pi`u astratte per l’implementazione del livello successivo. Lo strato di base (livello 0) consiste nell’hardware, mentre il pi`u alto rappresenta l’interfaccia utente. Secondo il principio della modularit`a, gli strati sono pensati in modo tale che ognuno utilizza funzionalit`a (operazioni) e servizi solamente degli strati inferiori.