

Chiamate di sistema

Normalmente il codice viene eseguito in un ambiente ristretto (*user mode*), con controlli effettuati a livello hardware; per effettuare alcune operazioni (MMIO, disabilitare interrupt, timer, modifica CPSR, ...) è necessario passare in *kernel mode*.

Questo sistema prende il nome di *dual mode operation*. La modalità in cui si trova il processore è memorizzata nella parola di stato (cioè insieme ai flag ZNCV), e il cambio avviene attraverso le chiamate di sistema.

Una syscall consiste nell'esecuzione di una funzione, scelta da un vettore configurato dal sistema operativo, in kernel mode tramite la generazione di uno specifico interrupt software.

- scrittura parametri e numero di syscall nei registri;
- interrupt software;
- individuazione dei parametri: registri e stack utente;
- copia dei parametri (anche valori puntati) da memoria utente a kernel (sicurezza);
- validazione dei parametri;
- esecuzione e scrittura dei risultati in memoria utente.

Tipicamente il codice utente chiama una funzione *stub* di libreria, che converte i parametri nella forma attesa dalla syscall, e a sua volta chiama una *kernel stub*. Questa effettua dei controlli sui parametri (non eseguiti nella funzione di libreria perché questa può essere bypassata da codice malevolo) e genera l'interrupt per l'esecuzione della chiamata.

È generalmente vantaggioso avere un numero limitato di syscall:

flessibilità è difficile fare modifiche al kernel (riavvio, problemi di compatibilità);

affidabilità un errore nel kernel può crashare l'intero sistema;

performance i context switch sono costosi, è preferibile implementare funzionalità in librerie che minimizzano le syscall.

Si delega alle syscall solo quello che non può essere implementato in codice utente.

In kernel mode le syscall sono solo `bl`.