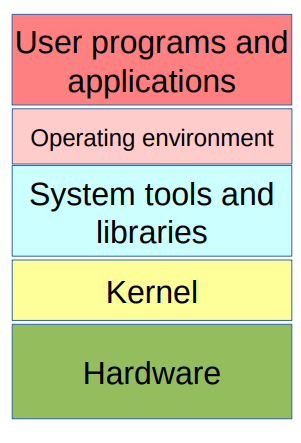
**Interazione processi-kernel**



Il kernel espone le funzionalità che implementa tramite un API specifica chiamata system call

Esiste uno standard che cercano di uniformare, POSIX, ma solo in fase di compilazione. Questo perché i kernel parlano lingue diverse.

Tramite strace si possono visualizzare le chiamate di sistema di un programma.

Come proteggere i servizi:

Il codice e il kernel condivide la stessa macchina, memoria e processore.

Un programma non può vedere la memoria di un altro

**Il contesto del kernel è separato da quello dei programmi**

**Autenticazione degli utenti e controllo degli accessi**

**Separare i contesti di esecuzione**

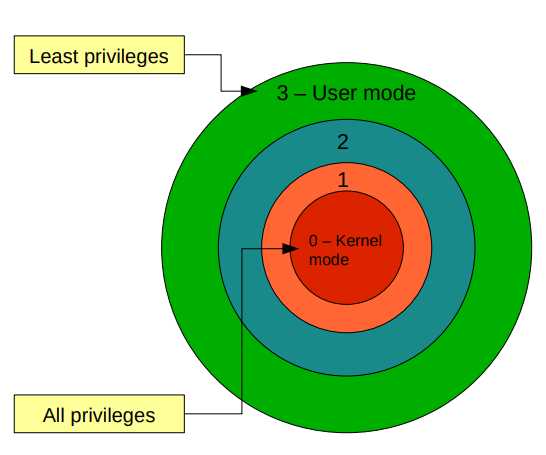
L'obbiettivo è quello di separare il kernel e i programmi in esecuzione, solo il kernel può lavorare sul hardware.

Privilegi lato kernel (privileged (kernel mode))

Ha completo controllo sull’hardware, ha una grossa responsabilità in caso il sistema vada in crash

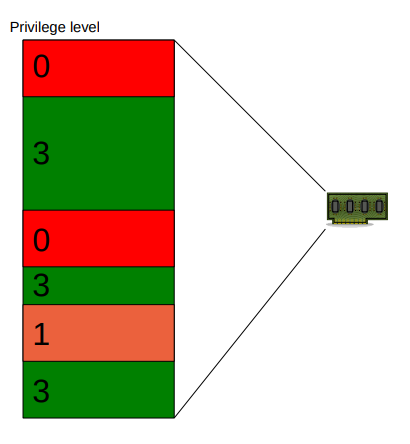
Privilegi lato utente(Unprivileged (user mode))

Controllo hardware tramite system call, gli errori non ha effetto su altri processi



Isolamento della memoria

Si mettono dei tag nelle aree di memoria.



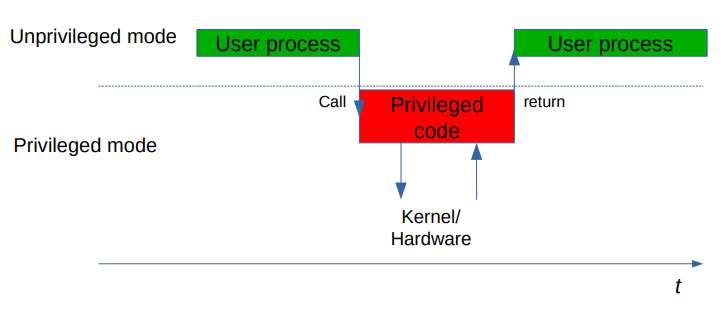
La macchina in fase di boot è privilegiata, ad esempio i driver.

Il codice lanciato da un utente root è comunque eseguito in modalità non-privilegiata.

La tipologia di un'istruzione (privilegiata o non-privilegiata) dipende anche dalla CPU.

Per creare funzionalità kernel bisogna creare dei moduli del kernel

**Comunicazione Kernel-Processo utente**



Le chiamate di sistema non sono uguali alle funzioni.

A livello hardware non è possibile saltare da un'area privilegiata a uno non privilegiata e viceversa-

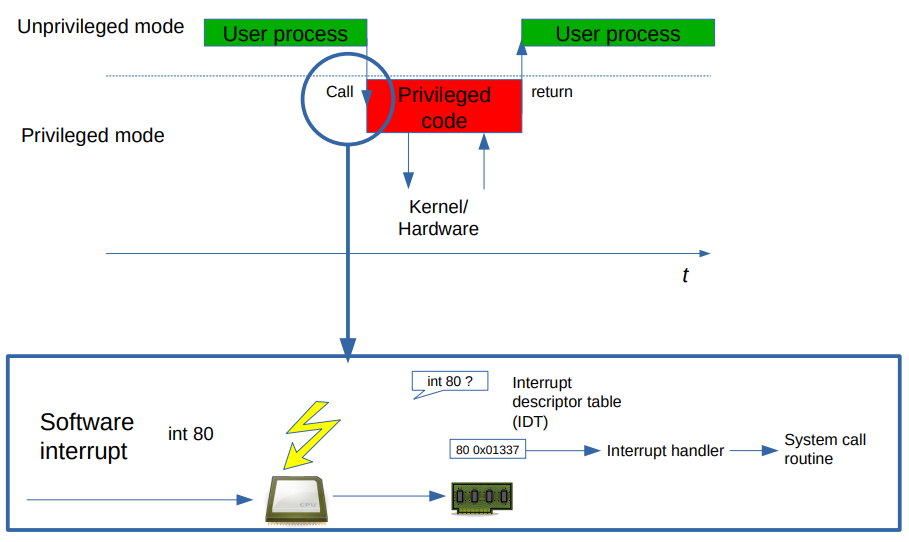
La soluzione a questa limitazione si può fare tramite dei gate.

I gate sono degli indirizzi specifici che consentono di saltare tra le due zone.

Gli interrupt interrompono l’esecuzione di un processo dando il controllo al OS.

Gli interrupt possono essere sia hardware che software.

Le chiamate di sistema sono degli interrupt software.



Int 80 è il codice di un interrupt software.

L'interrupt è lento, deve salvare il contesto, eseguire, ripristinare il contesto

Principio del privilegio minimo

Il programma deve avere solo i privilegi che necessita, usando solo le risorse necessarie

Il kernel ha diversi design:

* Monolitico: è un software unico che viene eseguito in modalità privilegiata
* Microkernel: solo una minima parte è gestita in modalità privilegiata
* Hybrid Kernel: alcune parti sono a livello kernel altre no

