

## SERVIZI DELL'SO >

L'S.O. mette a disposizione:

- User Interface: Di tipo CL (shell), GUI o Touch
- Program Execution:

Gestisce le risorse per l'esecuzione dei programmi

- I/O Operations:

Gestisce gli I/O richiesti dall'utente (processo).

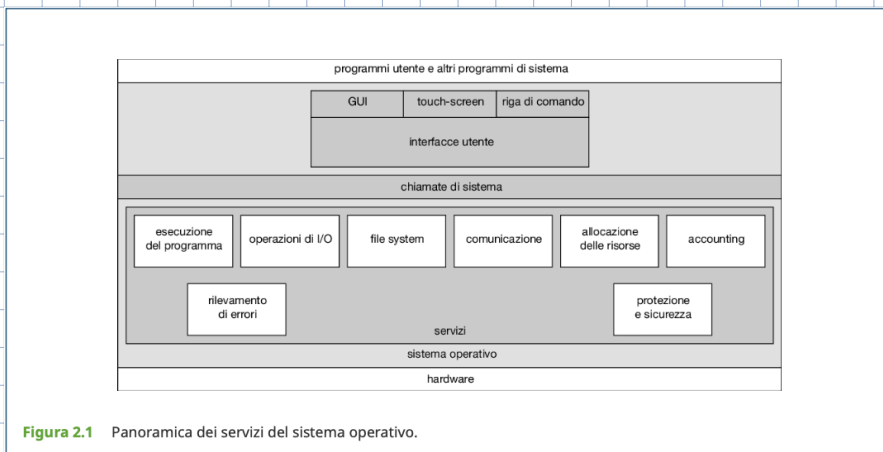
- Gestione della Memoria e File System
- Comunicazione: con altri sistemi
- Rilascamento degli Errori:

Sia nel software che nella gestione dell'hardware

- Allocazione delle risorse: per gestire più programmi
- Accounting:

Monitoraggio delle risorse e degli usi degli utenti

- Protection & Security



I servizi sono divisi in:

- Funz utili all'utente: Process execution, Communication, FS, I/O, Error handling, GUI.
- Funz. di efficientamento SO: Resource alloc, Accounting, Protection & security.

A seconda del tipo di SO tutti o alcuni servizi saranno nel kernel:

- Kernel "monolitico": tutti i servizi sono sul kernel
- micro-kernel: pochi = = = =

## SYSTEM CALLS >

Sono delle Application Programming Interfaces che consentono l'accesso ai servizi dell'S.O.

Ogni S.O. ha un tipo di API diversa:

- Windows: Win32
- Linux/MacOS: Posix
- SVM: Java API

Solitamente vengono definite in un linguaggio di alto livello (C).

### libc / stdio / C++

Sono librerie di interfaccia alle chiamate di sistema

### PASSAGGIO DI PARAMETRI

Possono essere passati così:

- registri
- blocchi di indirizzi di mem. nei registri
- stack: Utile quando si sfiora

### COMUNICAZIONE FRA PROCESSI

Le system call sono utilizzate anche nella comunicazione fra processi:

→ Message Passing Model:

Tramite un protocollo si scambiano messaggi

→ Shared Memory Model:

I proc. comunicanti hanno un'area di mem. condivisa per poter "depositare" / prelevare messaggi.

Richiede un processo di Sincronizzazione.

