

Appunti di Sistemi Operativi

Riccardo Lo Iacono

Dipartimento di Matematica & Informatica
Università degli studi di Palermo
Sicilia
a.a. 2022-2023

Indice.

1	Introduzione	2
1.1	SO e applicativi	2
2	SO e sue caratteristiche	3
2.1	GUI	3
2.2	Struttura	3

– 1 – Introduzione.

Un sistema operativo (SO) può essere pensato come un insieme di programmi, cui scopo è quello di gestire l'hardware e di interfacciare l'utente e la macchina.

Se si pensa ad un generico calcolatore, si può pensare a questi come l'insieme di *SO*, *hardware*, *applicativi*, *utenti*. Da tale decomposizione, il sistema operativo ha anche il compito di gestire il coordinamento dei vari applicativi.

Definizione: Un sistema operativo è l'unico programma sempre in esecuzione su una macchina con lo scopo di garantire il corretto funzionamento della stessa.

– 1.1 – SO e applicativi.

Si consideri un programma che è in esecuzione dalla CPU; ogni volta che questi necessita di un input dall'utente, o il verificarsi di un qualsiasi evento, questi genera un'eccezione. Si pensi alle eccezioni come dei messaggi alla CPU, la quale, terminata l'esecuzione dell'istruzione in corso, procede a risolvere.

Le eccezioni, anche dette interruzioni, sono di tre categorie *trap*, *fault*, *abort*.

- **Trap:** si tratta di eccezioni che sospendo l'istruzione, dopo la loro gestione riprendono l'esecuzione.
- **Fault:** si tratta di interruzioni che interrompono l'istruzione, dopo la gestione procederanno al riavvio della stessa.
- **Abort:** si tratta di eccezioni che interrompono l'istruzione, dopo la gestione riavvieranno il programma cui l'istruzione appartiene.

Ulteriore caratteristica delle interruzioni è il loro essere *sincrone*: cioè causate dalla stessa CPU; *asincrone*: se generate da un applicativo. Infine le interruzioni sono *mascherabili*: se è possibile posticipare la loro gestione, senza causare danni al sistema; *non mascherabili* viceversa.

– 2 – SO e sue caratteristiche.

Tra i vari compiti di un SO vi è anche quello di offrire servizi, servizi quali ad esempio la GUI: i dettagli relativi quest'ultima sono analizzati nel seguente paragrafo.

– 2.1 – GUI.

Si pensi alla GUI come una versione semplificata della CLI¹. In quanto versione semplificata risulta ovvio che la GUI sia meno potente della CLI, seppur svolgendone i medesimi compiti.

Si può in tal senso pensare alla GUI come un programma che ne esegue altri. Analizzando la generica esecuzione di un programma da GUI, questa fa sì che vengano eseguite, nell'ordine le *syscall* fork ed exec.

– 2.1.1 – SO e syscall.

Considerando una generica syscall, questa si compone di due parti quali un'identificatore e un insieme di parametri.

- **Identificatore:** codice univoco che permette di accedere in maniera diretta al programma.
- **Parametri:** insieme di informazioni utili all'esecuzione del programma. Si osservi che qual'ora il numero di parametri superi quello dei registri, i primi verranno memorizzati in un blocco di memoria, successivamente si passerà a parametro il puntatore a tale locazione.

– 2.2 – Struttura.

La struttura con cui un sistema operativo può essere realizzata non è univoca, esistono difatti tre tipologie principali di SO.

- **SO a struttura semplice:** anche detti monolitici, sono sistemi *tightly-coupled* in cui non vi è una netta suddivisione del sistema. Con questa tipologia, le applicazioni accedono direttamente alle routine, scrivendo direttamente su disco e/o video.
- **SO con struttura a strati:** sono sistemi *loosely-coupled* in cui il sistema è diviso in strati, ciascuno dei quali può utilizzare funzioni proprie o dello strato successivo.
- **SO con struttura a micro-kernel:** sono sistemi simili a quelli a strati, ne differiscono per i compiti assegnati al kernel: si fa sì che quest'ultimo gestisca solamente i compiti strettamente necessari alla sopravvivenza del sistema stesso.