

Organizzazione del kernel

Leonardo Bizzoni

January 30, 2024

Il kernel di un OS general-purpose è un programma di elevate dimensioni e complessità che deve operare rapidamente ed il cui malfunzionamento potrebbe causare un crash dell'intero sistema di elaborazione.

1 Struttura monolitica

Struttura utilizzata dall'originale sistema operativo Unix. Il kernel è composto da un singolo file binario statico, che fornisce tutte le funzionalità necessarie ai programmi applicativi ed è sempre in kernel-mode.

1.1 Vantaggi

- Elevate prestazioni.

1.2 Svantaggi

- Un bug in un servizio può causare il crash dell'intero sistema.
- Necessità di ricompilare per aggiungere funzionalità.

2 Struttura a strati

Lo strato più basso(*primo strato*) interagisce con l'hardware. L'*n*-esimo strato può interagire solo con l'*(n-1)*-esimo strato.

2.1 Vantaggi

- Ogni strato viene progettato ed implementato indipendentemente dagli altri.

- È facile testare la correttezza di uno strato indipendentemente dagli altri.
- Riduzione della complessità grazie all'astrazione.

2.2 Svantaggi

- È difficile definire quali funzionalità appartengono a quale strato.
- Ogni strato introduce un overhead che peggiora le performance.

3 Struttura microkernel

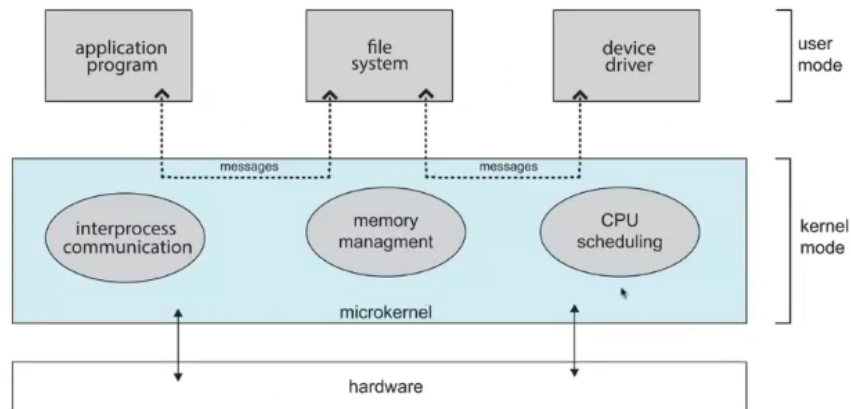
Questa struttura sposta quanti più servizi possibile fuori dal kernel in **programmi di sistema**, mantenendo l'insieme minimo di servizi indispensabili per implementare gli altri. Questo porta a dimensioni molto ridotte.

I servizi tipicamente offerti da un microkernel sono:

- scheduling dei processi
- gestione delle memoria
- comunicazione tra processi

Altri servizi come FS e device drivers vengono implementati a livello utente.

Per richiedere una funzionalità implementata da un programma di sistema, un'applicazione comunica con il programma che implementa tale funzionalità tramite le primitive di comunicazione offerte dal microkernel.



3.1 Vantaggi

- Facilità di estensione dell'OS. È possibile aggiungere nuovi servizi senza dover modificare direttamente il kernel.
- Facile controllare la correttezza del kernel.
- Maggiore affidabilità: se un servizio va in crash questo non viene propagato al kernel.

3.2 Svantaggi

- Maggiori overhead: la richiesta di un servizio deve transitare dal processo richiedente al kernel e dal kernel al programma di sistema destinatario che a sua volta può dover comunicare con un altro programma di sistema.

4 Struttura a moduli

Approccio con caratteristiche comuni con kernel stratificati e microkernel. Il kernel è strutturato in componenti **dinamicamente caricabili**, detti **moduli**, che comunicano tra di loro attraverso **interfacce**. Quando viene richiesto un determinato servizio viene caricato il modulo che lo implementa e quando questo non è più necessario viene scaricato.

I moduli vengono eseguiti in kernel-mode riducendo quindi l'overhead ma anche con minore isolamento tra di loro.

5 Struttura ibrida

In pratica pochi sistemi operativi adottano una struttura *pura*, quasi tutti combinano i diversi approcci allo scopo di ottenere performance, sicurezza e usabilità migliore.

Linux e Solaris, per esempio, sono kernel monolitici che supportano l'utilizzo di moduli, questo porta a performance elevate e, grazie ai moduli, dimensioni inferiori.

Windows era inizialmente un microkernel, successivamente diversi servizi sono stati inclusi nel kernel ed è stato aggiunto il supporto dei moduli.