



UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

Scuola di Scienze Matematiche, Fisiche e Naturali  
Corso di Laurea in Informatica

Tesi di Laurea

TITOLO ITALIANO

TITOLO INGLESE

NOME CANDIDATO

Relatore: *Relatore*  
Correlatore: *Correlatore*

Anno Accademico 2014-2015

Nome Candidato: *Titolo italiano*, Corso di Laurea in Informatica, © Anno  
Accademico 2014-2015

---

INDICE

---

1	introduzione a seL4A	7
1.1	Microkernel e kernel monolitici . . . . .	8
1.2	seL4 . . . . .	9



---

## ELENCO DELLE FIGURE

---



*"Inserire citazione"*  
— *Inserire autore citazione*





---

## INTRODUZIONE A SEL<sub>4</sub>A

---

Un sistema operativo (SO) è un insieme di software che gestisce le risorse hardware e software di un sistema di elaborazione fornendo servizi agli applicativi utente.

In un computer quindi il sistema operativo fornisce l'unica interfaccia diretta con l'hardware e in quanto tale ha un accesso esclusivo con il massimo dei privilegi detto *kernel mode*. Questo comporta che una vulnerabilità all'interno del sistema operativo può portare a gravi conseguenze per l'integrità e la sicurezza del sistema, inoltre qualche malintenzionato potrebbe approfittare di questo bug per trarne profitto. Uno degli obiettivi principali di un SO è quindi quello di garantire la sicurezza, ulteriore scopo è l'efficienza: un buon sistema operativo deve saper sfruttare al meglio tutte le risorse che ha a disposizione, dalla gestione della memoria per sfruttare al meglio lo spazio alla schedulazione dei processi per ottimizzare i tempi di esecuzione. Come ultimo obiettivo, ma non per questo meno rilevante, deve rendere il più semplice possibile l'utilizzo del dispositivo su cui è installato. Dalla definizione di SO data a inizio capitolo possiamo isolare una specifica parte di codice che è quella che permette al software di interfacciarsi con l'hardware, quindi l'accesso e la gestione delle risorse di un dispositivo, questa specifica parte si chiama *kernel* che come suggerisci il nome (nocciolo dall'inglese) rappresenta la parte centrale di un sistema operativo su cui tutto il resto si appoggia.

## 1.1 MICROKERNEL E KERNEL MONOLITICI

Esistono vari modelli strutturali per i sistemi operativi: monolitici, modulari, a livelli, microkernel ed ibridi, ad oggi i più diffusi sono gli ibridi, che combinano i vari modelli tra di loro, ma che in gran parte si basano su sistemi monolitici i quali consistono di un unico file binario statico al cui interno sono definite tutte le funzionalità del kernel e che viene eseguito in un unico spazio di indirizzi, questo comporta dei vantaggi e degli svantaggi:

- efficienza → motivo principale per cui la maggior parte dei sistemi operativi ancora si basano su kernel in gran parte monolitici, lavorando nello stesso spazio di indirizzi e gestendo tutto attraverso chiamate di sistema il SO risulterà molto reattivo e performante
- semplicità → in quanto non ha una vera e propria strutturazione ma il codice è tutto in un unico file binario risulta chiaramente più semplice da sviluppare.

D'altra parte ha anche degli svantaggi:

- inserimento di un nuovo servizio → questo richiede la ricompilazione del kernel, quindi non permette l'inserimento di un nuovo servizio a runtime
- dimensione → dovendo gestire tutte le principali funzionalità del sistema operativo il kernel sarà composto da milioni di righe di codice (MLOC - linux ha circa 20MLOC) e questo porta direttamente al successivo grosso svantaggio
- bug → maggiore è il numero di righe di codice maggiore sarà il numero di possibili bug, essendo tutto il codice eseguito nello stesso spazio di indirizzi un bug rischia di far bloccare l'intero sistema anche se il problema fosse molto piccolo e isolato a una minima funzione del kernel.

All'estremo opposto troviamo i *microkernel* che sono composti da un kernel ridotto al minimo indispensabile mentre tutto il resto deve essere gestito da server che operano sopra al microkernel, quindi in spazi di indirizzi separati.

Magari argomentare menzionando anche le politiche e i meccanismi, pro e contro di entrambi, menzionare anche il fatto che nei microkernel viene quindi ridotta la parte di codice eseguita con i massimi privilegi.

## 1.2 SEL4

seL4 fa parte della famiglia dei microkernel L4 che risalgono alla prima metà degli anni '90 creato da Jochen Liedtke per sopperire alle scarse performance dei primi sistemi operativi basati su microkernel.



---

## BIBLIOGRAFIA

---

- [1] Gernot Heiser. The sel4 microkernel - an introduction. *The seL4 Foundation*, Revision 1.2, 2020.
- [2] Wikipedia, l'enciclopedia libera. Kernel, 2023. ultima modifica 7 giu 2023.
- [3] Wikipedia, l'enciclopedia libera. Operating system, 2023. ultima modifica 16 luglio 2023.
- [4] Rosario Pugliese. Chiedere al prof come citare le sue slide.