

# Analisi delle performance del Linux Kernel Runtime Guardian

---

Presentata da Simone Magnani

Relatore Gabriele D'Angelo

18 Ottobre 2018

Alma Mater Studiorum - Università di Bologna

Campus di Cesena

Scuola di Scienze

Corso di laurea in Ingegneria e Scienze Informatiche

# Indice dei contenuti

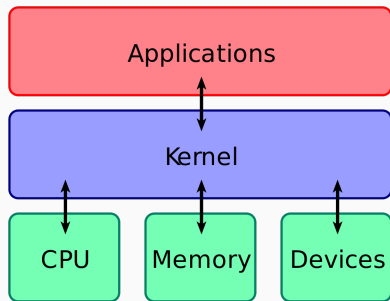
- Introduzione al sistema operativo
- Linux Kernel Runtime Guardian
- SysBench
- Analisi dei risultati ottenuti

# Introduzione al sistema operativo

---

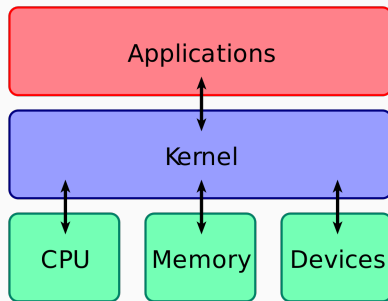
# Kernel Linux

- Layer d'astrazione per memoria, CPU e periferiche
- Monolitico
- Composto da moduli



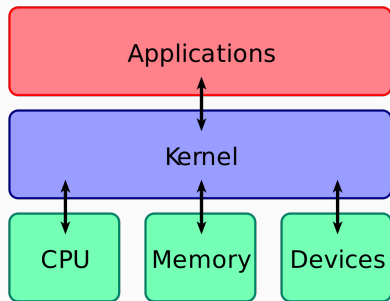
# Kernel Linux

- Layer d'astrazione per memoria, CPU e periferiche
- Monolitico
- Composto da moduli



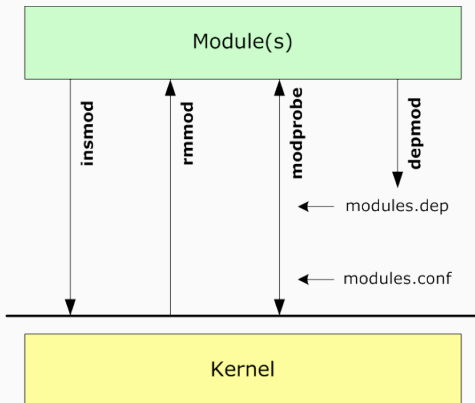
# Kernel Linux

- Layer d'astrazione per memoria, CPU e periferiche
- Monolitico
- Composto da moduli



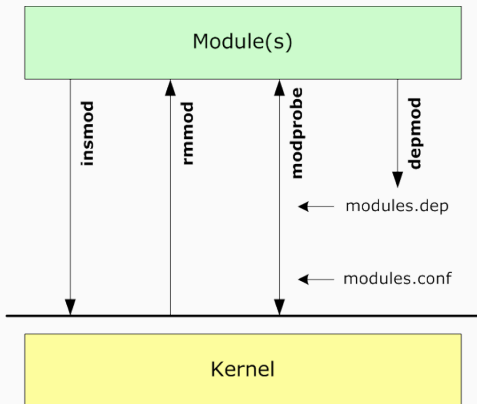
# I moduli del kernel

- File oggetto caricati e rimossi a runtime
- Estendono le funzionalità del kernel



# I moduli del kernel

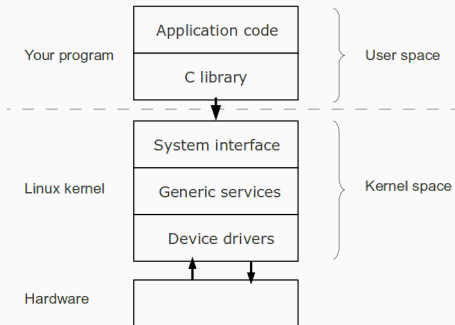
- File oggetto caricati e rimossi a runtime
- Estendono le funzionalità del kernel





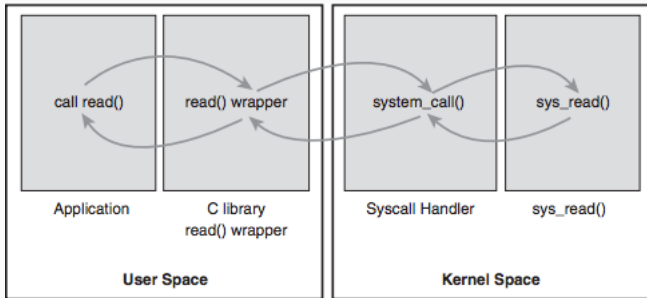
# Spazi d'esecuzione

## Kernel vs user space



# System call

Modalità per interagire con il sistema operativo



# Linux Kernel Runtime Guardian

---

# LKRG - Introduzione

È un modulo con l'obiettivo di eseguire controlli d'integrità del kernel e rilevare possibili attacchi di tipo 'exploitation'



# LKRG - Funzionamento

- Salvataggio e confronto degli hash di alcune regioni del sistema
- Monitoraggio di alcune system call
- Routine di validazione a seconda degli eventi scatenati



# LKRG - Funzionamento

- Salvataggio e confronto degli hash di alcune regioni del sistema
- Monitoraggio di alcune system call
- Routine di validazione a seconda degli eventi scatenati



# LKRG - Funzionamento

- Salvataggio e confronto degli hash di alcune regioni del sistema
- Monitoraggio di alcune system call
- Routine di validazione a seconda degli eventi scatenati



# SysBench

---

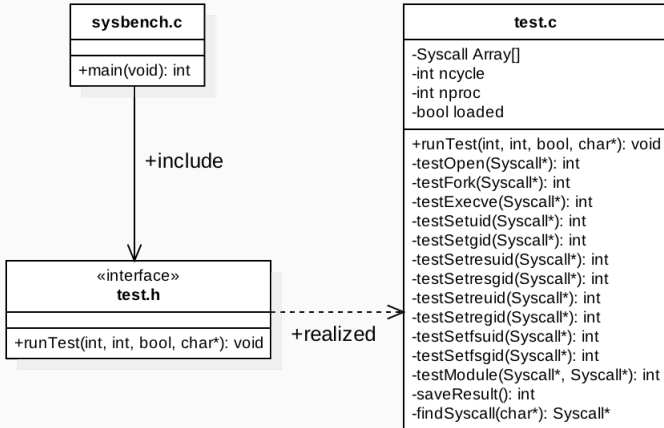


# SysBench - Obiettivo

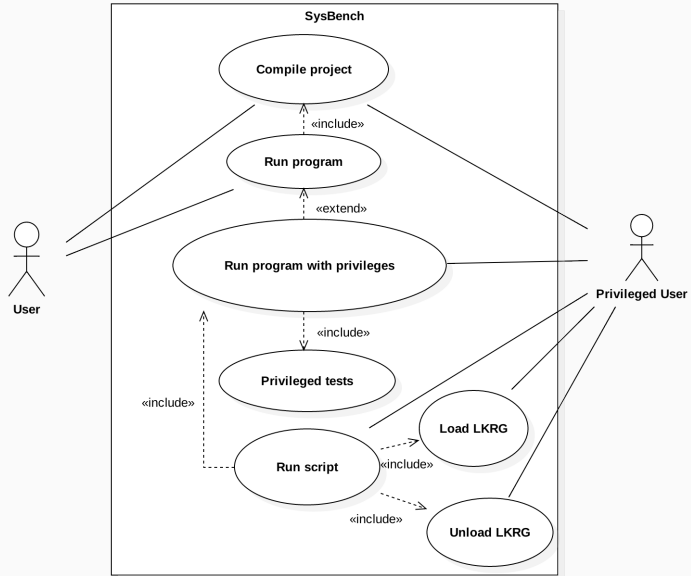
Software scritto in linguaggio C al fine di misurare il tempo d'esecuzione di alcune system call monitorate dal Linux Kernel Runtime Guardian



# SysBench - Architettura



# SysBench - Caso d'uso



## Analisi dei risultati ottenuti

---

# Analisi - Informazioni dei sistemi

Host macOS High Sierra



Guests Linux



Ubuntu

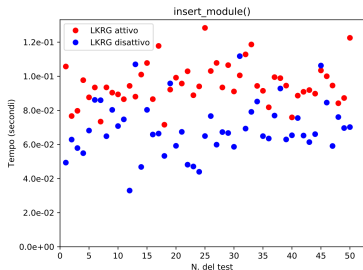
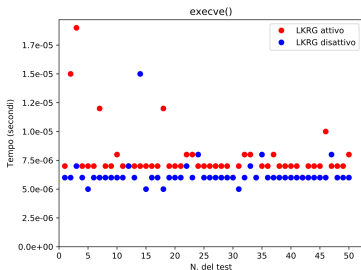
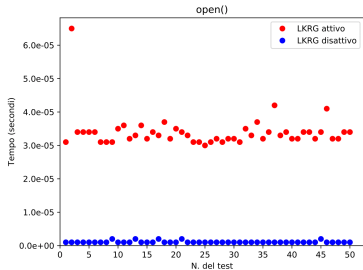
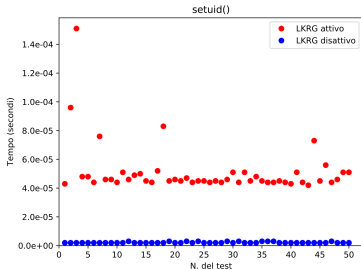


Debian

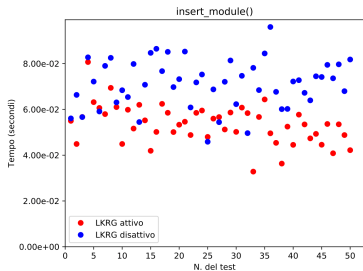
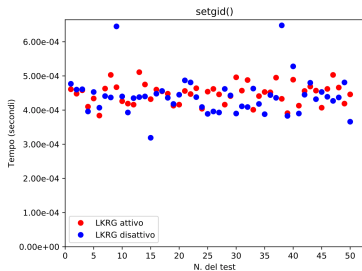
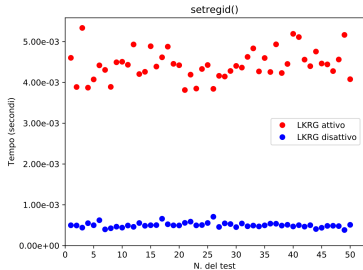
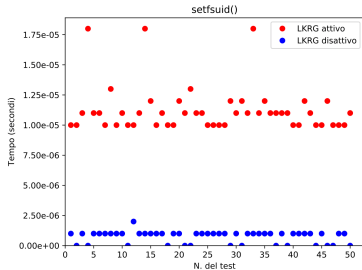


Mint

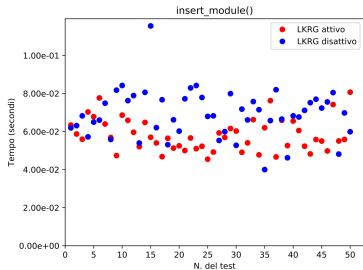
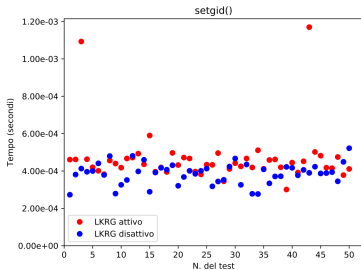
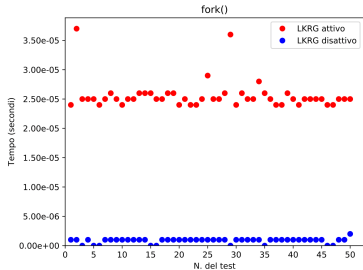
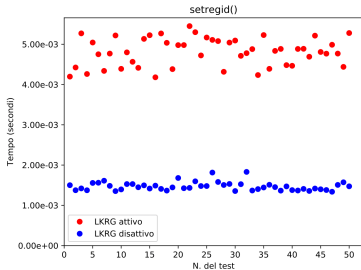
# Analisi - Test in Ubuntu



# Analisi - Test in Debian



# Analisi - Test in Mint



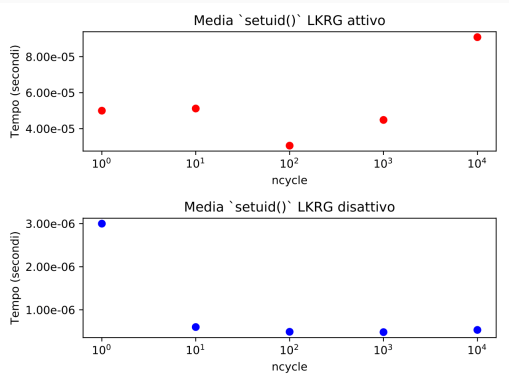


# Analisi - Sistemi a confronto

SystemCall	$\Delta$ Ubuntu (%)	$\Delta$ Debian (%)	$\Delta$ Mint (%)
setuid()	2332	2193	1875
setgid()	213	102	121
setresuid()	3659	735	2328
setresgid()	296	1597	365
setreuid()	3144	1478	2136
setregid()	283	884	327
setfsuid()	3406	1519	3347
setfsgid()	3824	1861	4746
open()	3089	1978	3051
fork()	2792	1906	3043
execve()	123	143	115
insert_module()	137	75	84
delete_module()	3432	1411	3581
Media	2056	1222	1932

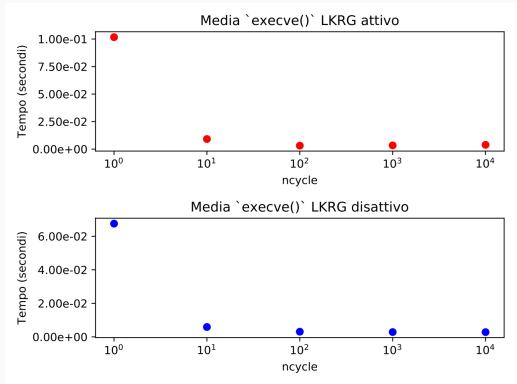
# Analisi - Ottimizzazioni

- Influenza rilevante di LKRG
- Influenza irrilevante di LKRG



# Analisi - Ottimizzazioni

- Influenza rilevante di LKRG
- Influenza irrilevante di LKRG



# Conclusioni

---

# Ringraziamenti

Un ringraziamento speciale alle seguenti persone:

- Gabriele D'Angelo
- Rosanna, Fabrizio e Francesco
- Catarina
- Tutti gli amici più cari
- CeSeNA Security Team



# FINE

A cura di Simone Magnani

( s41m0n )

< simonemagnani.96@gmail.com >

