

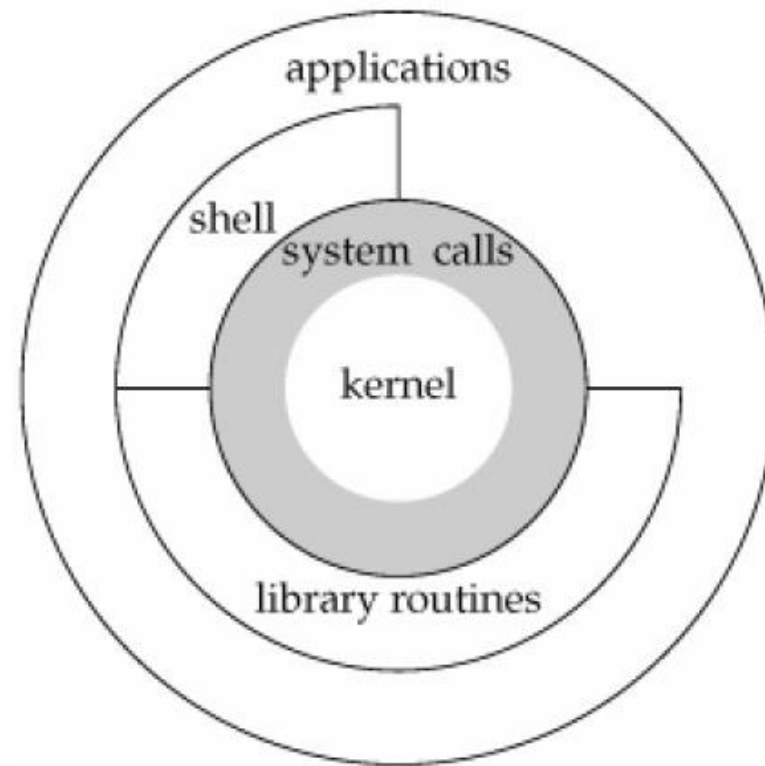
The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on this are several faint, white geometric elements: concentric circles, arcs, and degree markings. A large arc on the left side is marked with degrees from 140 to 260 in increments of 10. Other smaller arcs and circles are scattered across the frame, some with arrows indicating direction.

SYSTEM CALLS

MATTEO BECCARI

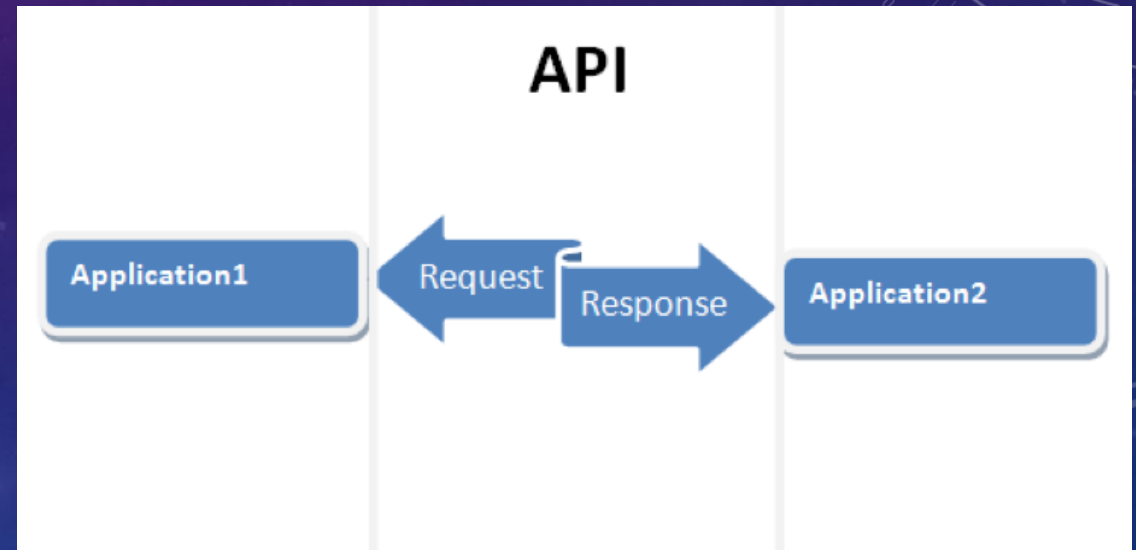
DEFINIZIONE

- Per utilizzare i servizi del sistema operativo Linux, ed interfacciarsi col kernel, il programmatore ha a disposizione una serie di “entrate” per il kernel chiamate System Calls.



API (APPLICATION PROGRAM INTERFACE)

- L'API collega le funzioni del sistema operativo ai programmi utente. Crea un collegamento tra il sistema operativo e un processo, consentendo ai programmi a livello utente di richiedere servizi del sistema operativo.



ESEMPIO

Il programma raffigurato stampa a schermo in quale core del processore è in esecuzione, tramite una chiamata di sistema

```
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <sys/syscall.h>
4  #include <sys/types.h>
5
6  int main()
7  {
8      unsigned cpu;
9      syscall(SYS_getcpu, &cpu, NULL);
10     printf("Questo programma è in esecuzione nel core: %u\n", cpu);
11     return 0;
12 }
13
```


SYSTEM CALLS PIU' USATE

- OPEN(): consente di accedere a un file su un file system.
- WRITE(): utilizzato per scrivere dati da un buffer utente a un dispositivo come un file.
- READ(): viene utilizzato per ottenere dati da un file nel file system.
- WAIT(): utilizzata per sospendere il processo padre.
- FORK(): i processi generano cloni di se stessi (processi figli)
- EXIT(): viene utilizzata per terminare l'esecuzione del programma.