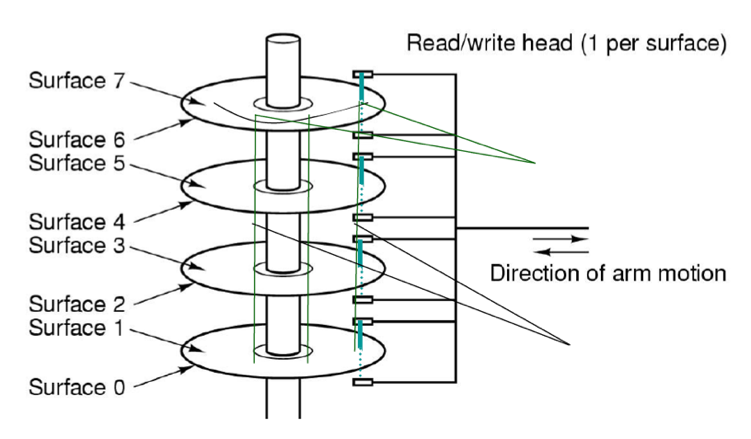
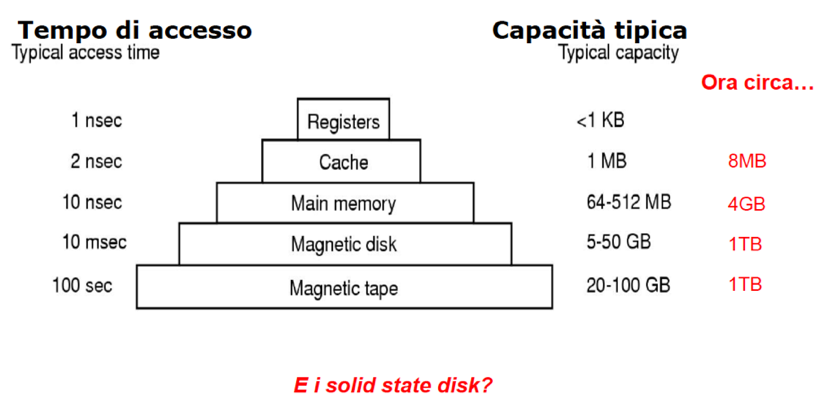
**Ricapitolazione di concetti di base**

**Gerarchia fisica di memoria**

I registri sono interni alla CPU; dimensione:

32 bit su processori a 32-bit

64 bit su processori a 64-bit

La cache è controllata da hw ed è suddivisa in blocchi chiamati linecon ampiezza tipica 64 B

* L1 dentro la CPU, L2 adiacente alla CPU (L2 condivisa (Intel) o propria di ciascun core (AMD)?)
* Hit (2 cicli di clock), miss(memoria)
* Write through, copy back

I dischi magnetici hanno capienza 100 volte superiore e costo/bit100 volte inferiore rispetto alla RAM, ma per tempo di accesso 1000 volte peggiore

**Programmi e Memoria**

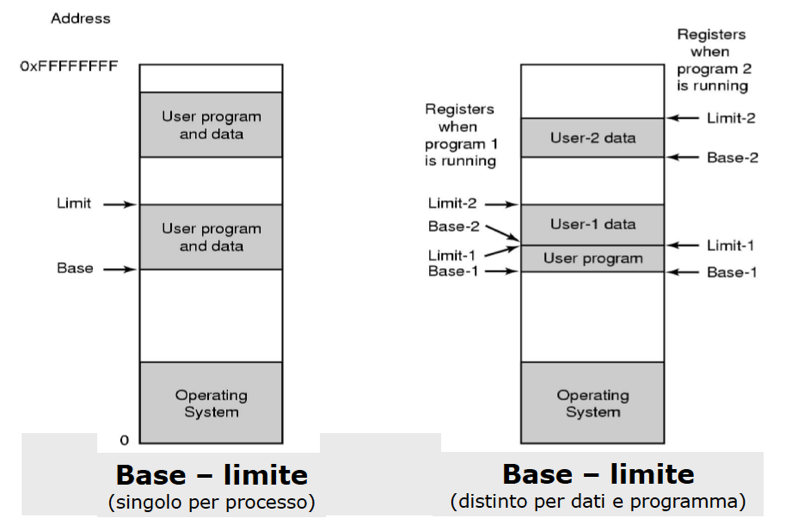
CMOS è una memoria volatile ma alimentata da una piccola batteria

* Memorizza ora e da quale disco fare boot
* Memorizza impostazioni BIOS diverse da default

Interazione memoria-programmi:

* Come proteggere I programmi tra loro e il kernel dai programmi
* Come gestire la rilocazione
* Quando si compila un programma non si sa in che area della memoria verrà caricato (indirizzi da 0 a …). Soluzione hardware, due registri (base e limite), verifica e somma base+indirizzo costa qualche ciclo di CPU

**Vista logica della RAM**

Nella sua forma più rudimentale la ripartizione della RAM tra processi distinti utilizza 2 registri speciali

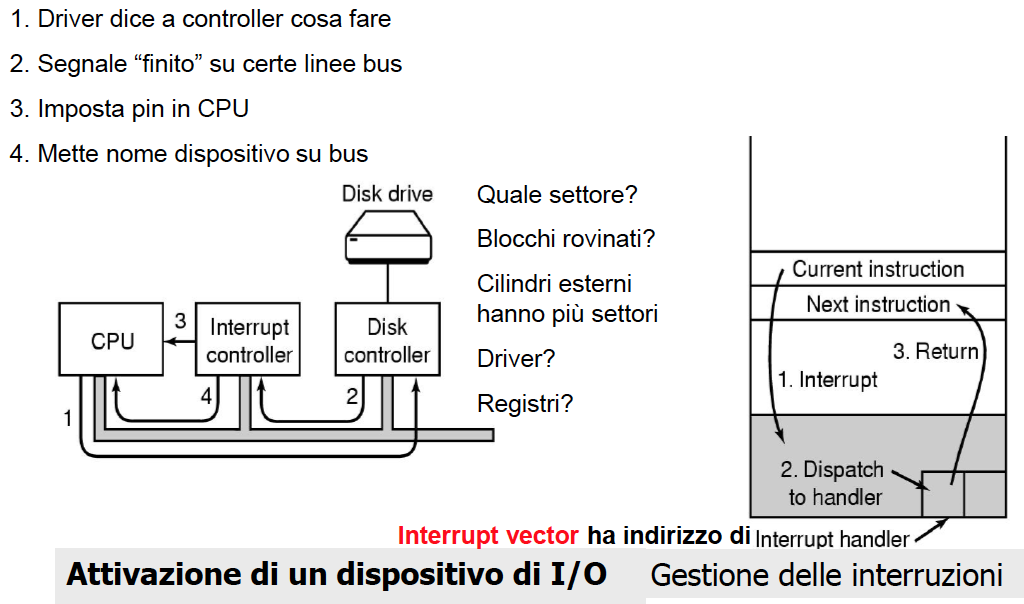
* Basee limite, i cui valori formano parte importante del contesto del processo
* L’allocazione del processo in RAM richiede rilocazionedella sua memoria virtuale

In generale la gestione dello spazio di memoria virtuale dei processi utilizza un dispositivo di MMU(Memory Management Unit) logicamente interposto tra CPU e memoria

* Sotto la responsabilità del Sistema Operativo

Attenzione: a ogni context switch la cache è piena di dati del processo precedente

**Trattamento delle interruzioni**



L’uso delle interruzioni per l’interazione con i dispositivi evita il ricorso al polling.

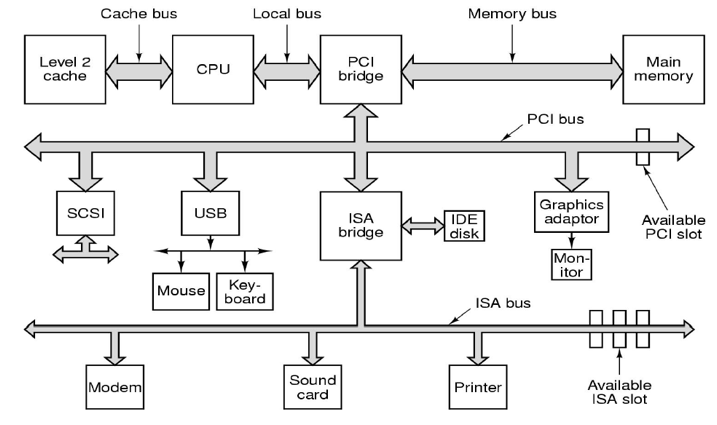
L’interazione tipica avviene in 4 passi successivi come illustrato in figura

1. Il gestore del dispositivo programma il controllore di dispositivo scrivendo nei suoi registri di interfaccia
2. Il controllore agisce sul dispositivo e poi informa il controllore delle interruzioni
3. Il controllore delle interruzioni asserisce un valore (pin) di notifica verso la CPU
4. Quando la CPU si dispone a ricevere la notifica il controllore delle interruzioni comunica anche l’identità del dispositivo (così che il trattamento dell’interruzione sia attribuito al gestore appropriato)

All’arrivo di una interruzione

* I registri PC (Program Counter) e PSW (Program Status Word) sono posti sullo stackdel processo corrente
* La CPU passa al “modo operativo protetto”
* Il parametro principale che denota l’interruzione serve come indice nel vettore delle interruzioni (così si individua il gestore designato a servire l’interruzione)
* La parte immediata del gestore esegue nel contesto del processo interrotto (la parte del servizio meno urgente può essere invece differita e demandata a un processo dedicato)

**Pentium: architettura fisica**



ISA bus(Industry Standard Architecture, 8.33 MHz, 2B/ciclo, 16.67 MB/s)

– Non più usato

SATA: I(1,5 Gbit/s), II(3 Gbit/s), III(6 Gbit/s)

PCIbus (Peripheral Component Interconnect, 66 MHz, 8 B/ciclo, 528 MB/s)

– Di vecchia concezione ma dotato di connettori per una grande varietà di dispositivi

– PCI Express: 1.x 4GBB/sec, 2.x 8GB/sec, 3.x 16GB/sec

Memorybus(100 Mhz, dedicated)

–1333Mhz è attualmente lo standard per le DDR3

USBbus(Universal Serial Bus)

–USB 1.0: 1,5 Mbit/s

–USB 1.1: 12 Mbit/s

–USB 2.0: 480 Mbit/s

–USB 3.0: 4,8 Gbit/s

–Per l’interconnessione di dispositivi lenti (tastiera, mouse), con 4 linee delle quali 2 di alimentazione del dispositivo

–Unico device driver

–Buscon singolo mastercentrale predefinito che interroga @ 1 ms i dispositivi collegati (collegabili)

SCSIbus(Small Computer System Interface, ≤ 160 MB/s)

–Per l’interconnessione di dispositivi veloci (dischi veloci, un tempo anche scanner)

**Plug & Play (Pray?)**

Intel e Microsoft

Prima ogni scheda I/O aveva un livello di interrupt fisso e un indirizzo fisso per i registri

–Se si acquistavano due dispositivi con lo stesso valore di interrupt?

Con Plug & Play, il sistema assegna centralmente i livelli di interrupt e gli indirizzi di I/O e poi li rivela alle schede

**BIOS**

BIOS (Basic Input Output System)

* Contiene sw a basso livello per la gestione di I/O
* Viene caricato all’avvio del computer
* Verifica quanta RAM e quali dispositivi base (tastiera,ecc) sono presenti
* Fa scan dei bus ISA e PCI per rilevare dispositivi ad essi connessi

• I dispositivi vecchi (prima di plug & play, detti legacy) sono rilevati e registrati

• Vengono registrati anche i dispositivi plug & play

• Se ci sono nuovi dispositivi dall’ultimo avvio, questi vengono configurati (assegnati livelli di interrupt e indirizzi I/O)

* Determina il dispositivo di boot dalla lista in memoria CMOS

**BOOT**

Il primo settore del dispositivo di boot viene letto in memoria ed eseguito

* Contiene un programma che esamina la tabella di partizione e determina quale partizione sia attiva
* Da tale tabella, viene caricato un secondo boot loader

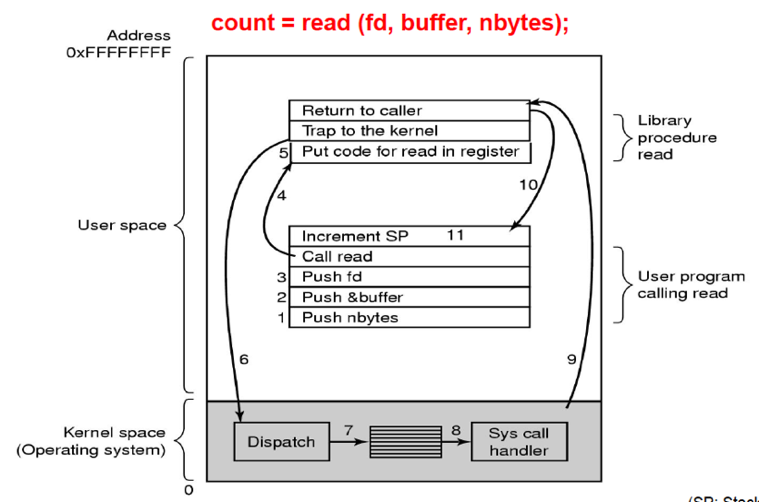
Legge il sistema operativo dalla partizione attiva e lo esegue

* Il sistema operativo interroga il BIOS per ottenere informazioni sulla configurazione del sistema

Per ogni dispositivo controlla l’esistenza del driver

* Se non c’è chiede di inserire CD o Floppy
* Se ci sono li carica nel kernel
* Poi, esegue varie inizializzazioni ed esegue programma iniziale (login, GUI)

**Chiamate di sistema**



La maggior parte dei servizi del Sistema Operativo sono eseguiti in risposta a invocazioni esplicite di processi (chiamata di sistema)

Le chiamate di sistema sono nascoste in procedure di libreria predefinite

* L’applicazione non effettua direttamente chiamate di sistema
* La procedura di libreria svolge il lavoro di preparazione necessario ad assicurare la corretta invocazione della chiamata di sistema

La prima istruzione di una chiamata di sistema (trap) deve attivare il modo operativo privilegiato

* Inizia esecuzione ad un indirizzo prefissato del kernel
* Il parametro della chiamata designa l’azione da svolgere e la convenzione per trovare gli altri eventuali parametri
* Il meccanismo complessivo è simile a quello già visto per il trattamento delle interruzioni

•Le interruzioni sono asincrone

•Le chiamate di sistema invece sono sincrone

1. Il programma applicativo effettua una chiamata di sistema

•(2.-3.) Prima pone sullo stacki parametri secondo una antica convenzione C/UNIX

4. Poi invoca la procedura di libreria corrispondente alla chiamata

5. Questa pone l’ID della chiamata in un luogo noto al S/O

6. Poi esegue l’istruzione trapper passare all’esecuzione in modo operativo privilegiato

7. Il S/O individua la chiamata da eseguire

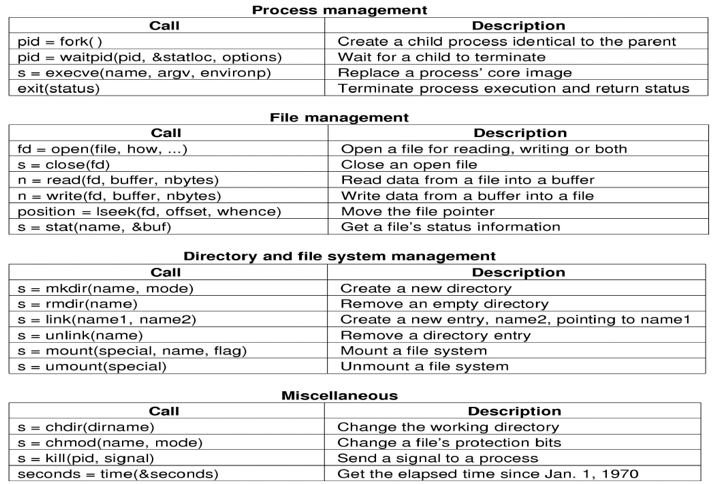
8. La esegue

9. Poi ritorna al chiamante oppure a un nuovo processo

10. Ritorna come farebbe da return di procedura

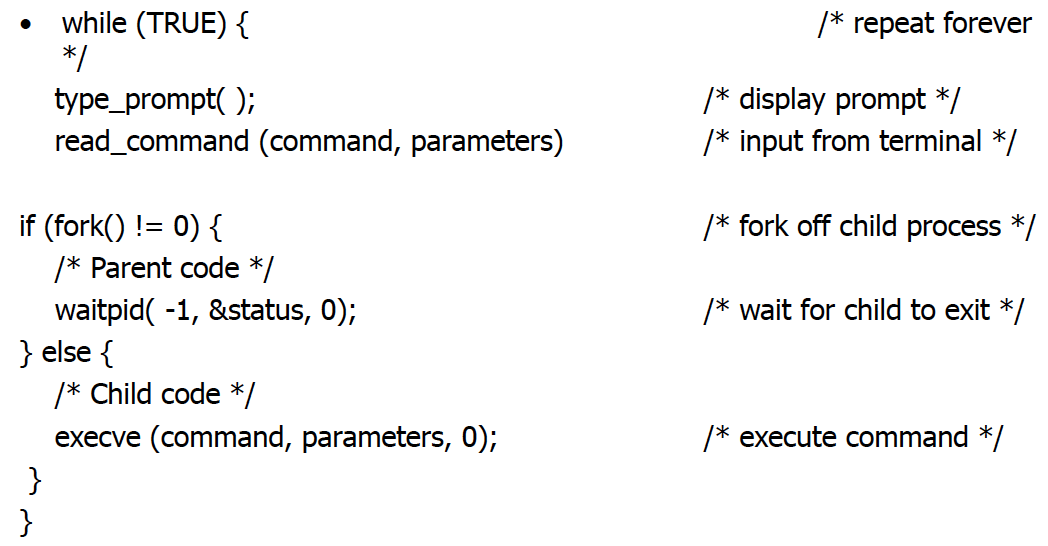
11. Cancella dati nello stack facendo avanzare il puntatore

**Alcune Chiamate di Sistema (POSIX)**

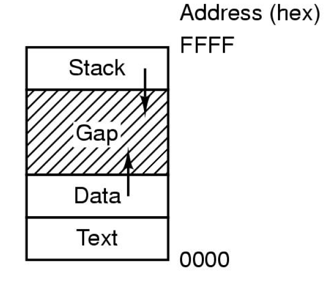
****

**System Calls (1)**

Una shell base tramite fork (UNIX):



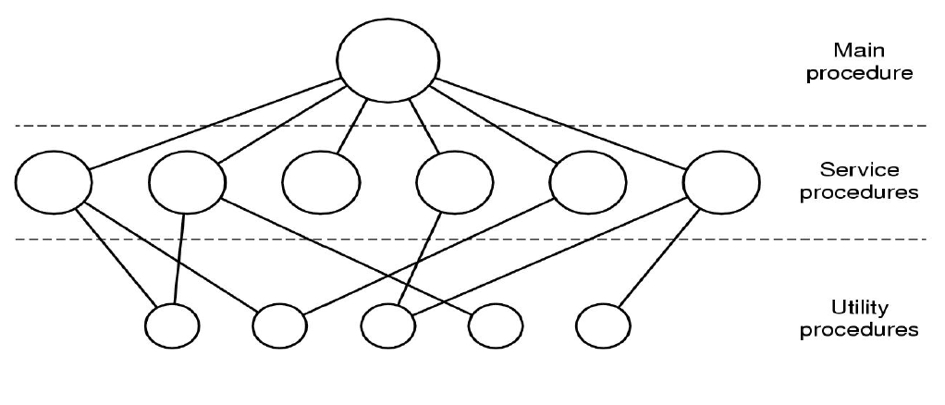
**System Calls (2)**



Processi in memoria hanno 3 segmenti: testo, dati, stack

**Architettura logica di S/O**

Struttura monolitica



Un’architettura monolitica non ha struttura

* Il S/O è una collezione “piatta” di procedure

•Ognuna delle quali può chiamarne qualunque altra

•Nessuna forma di information hiding

* Il S/O è un singolo .o che collega tutte le procedure che lo compongono

L’unica struttura riconoscibile in essa è data dalla convenzione di attivazione delle chiamate di sistema – Parametri messi in un posto preciso (stack) e poi esegue trap

• Organizzazione di base:

1. Programma principale che invoca le procedure di servizio richieste;

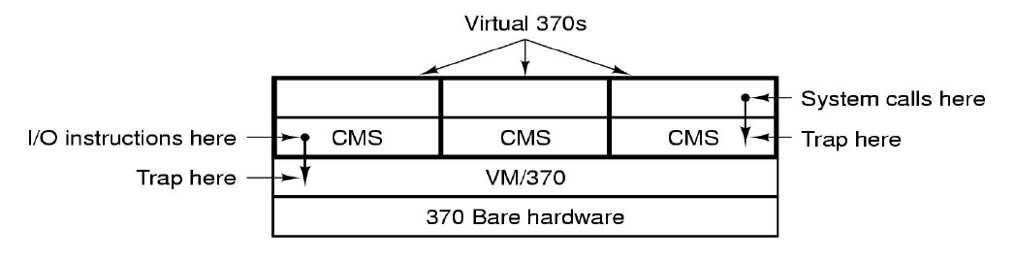
2. Insieme di procedure che eseguono le system call;

3. Insieme di procedure di utilità che sono di ausilio per le procedure di servizio.

• Generalizzazione è la struttura a strati (layers)

–THE system (Dijkstra ‘68)

Sistema a macchina virtuale



Primi anni ’70

- VM/370sviluppato da IBM Scientific Center, Cambridge, Massachussets per fornire time sharing su sistemi batch dell’azienda

- Basato sull’intuizione che un S/O a divisione di tempo in realtà realizza 2 fondamentali funzioni

1.Multiprogrammazione

2.Virtualizzazione dell’elaboratore fisico

- Separandole e ponendo 2. alla base si possono offrire copie identiche di “macchine virtuali” (copie dell’hardware) a S/O distintiche realizzano 1. secondo un criterio loro proprio

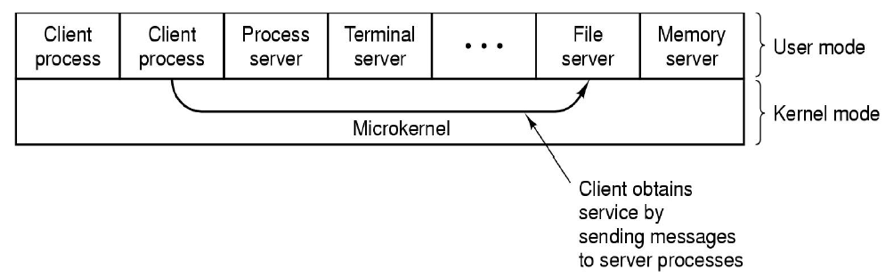
CMS(Conversational Monitor System)

* S/O interattivo a divisione di tempo mono-utente
* Esegue sopra una macchina virtuale realizzata da VM/370

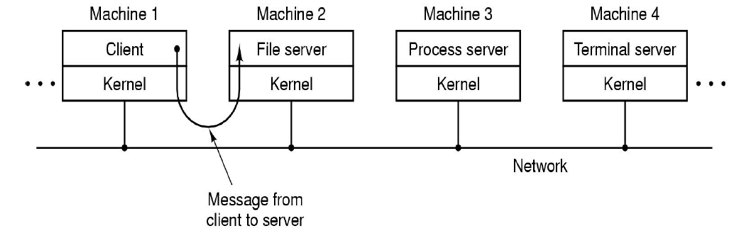
L’idea della virtualizzazione di elaboratori logici o fisici ha avuto notevole seguito

* Intel: modo 8086 virtuale su Pentium
* MS Windows & co.: ambiente virtuale di esecuzione MS-DOS
* JVM: architettura portabile di elaboratore logico

Struttura di tipo cliente-servente



Struttura distribuita



L’architettura di S/O a modello cliente-servente è anche detta a micro-kernel

L’idea portante è di limitare al solo essenziale le responsabilità del nucleo delegando le altre a processi di sistema nello spazio di utente

* I processi di sistema sono visti come serventi
* I processi utenti sono visti come clienti

Il ruolo del nucleo di S/O è di gestire i processi e supportare le loro comunicazioni

Se un servizio va in crash difficilmente lo farà tutto il sistema

Idea “pulita” ma prestazioni generalmente scadenti

**Monolitico vs. Microkernel**

http://www.dina.dk/~abraham/Linus\_vs\_Tanenbaum.html

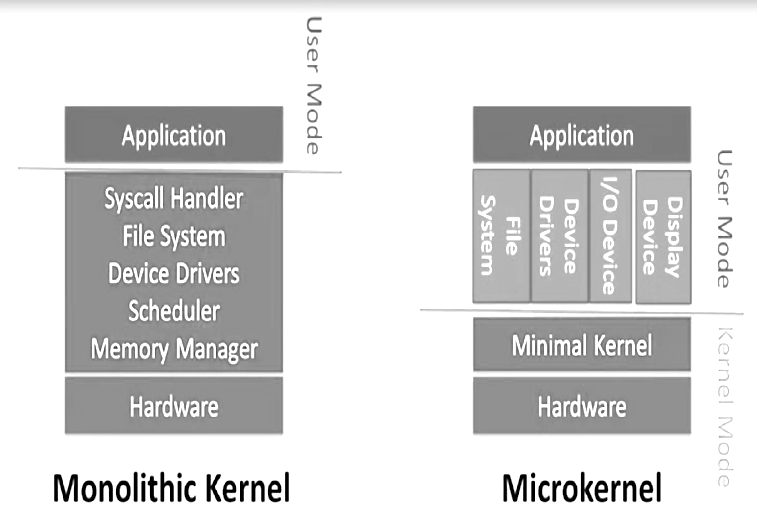
I kernel monolitici sono più semplici da realizzare e mantenere.

I microkernel consentono gestione più flessibile

* montare un certo servizio (es. un hard disk) solo per un utente invece che per tutti, il “montaggio” diventa un’operazione livello utente, non più livello kernel

I microkernel hanno problemi di sincronizzazione tra le varie componenti, che ne rallentano sviluppo e mantenimento

Linux è Monolitico, Minix è Microkernel, Windows e Mac OS/X sono ibridi.



**Unità metriche**

