

Università di Roma Tor Vergata
Corso di Laurea triennale in Informatica
Sistemi operativi e reti
A.A. 2017-18

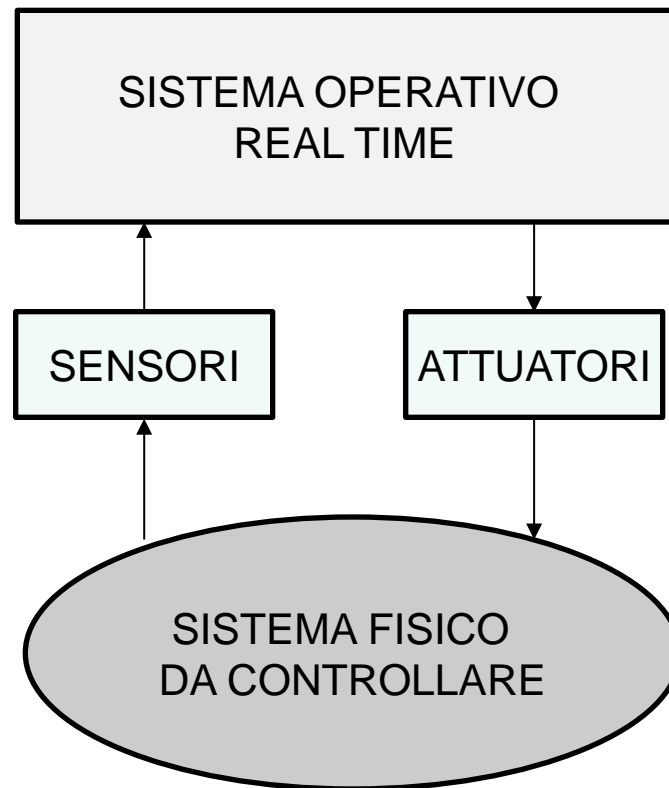
Pietro Frasca

Lezione 3

Martedì 10-10-2017

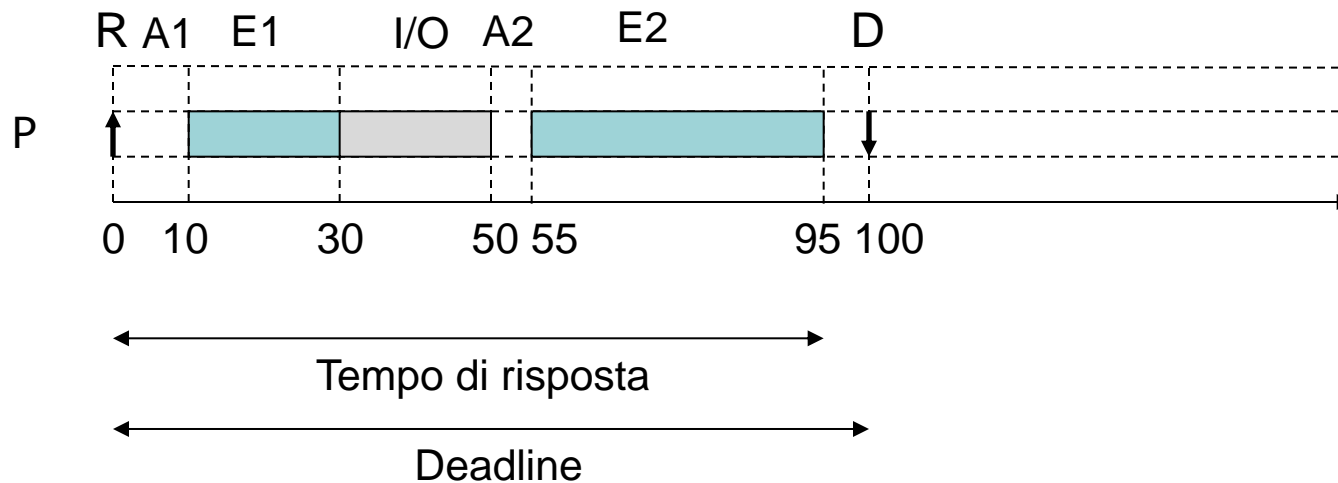
Sistemi operativi real time

- I *SO real time (tempo reale)* costituiscono un'altra classe di sistemi operativi che usa la multiprogrammazione.
- Il calcolatore è utilizzato per la gestione e il controllo di un sistema fisico, detto ***ambiente operativo***, costituito ad esempio da impianti industriali, centrali elettriche, robot, etc.



- Caratteristica fondamentale dei SO realtime è che ogni task deve essere eseguito entro un intervallo di ***tempo definito (deadline)*** imposto dall'applicazione (da microsecondi a millisecondi).
- In altri termini, la validità dei risultati ottenuti da un programma non dipende solo dalla correttezza del programma, ma anche dall'intervallo di tempo entro il quale i risultati sono prodotti.
- Ad esempio, supponiamo che un'applicazione real time, quando avviata, esegua dapprima un cpu burst di 20 ms, quindi I/O burst per altri 20 ms e infine di nuovo cpu burst per 40 ms. Supponiamo, inoltre, che tale programma sia stato progettato affinché esegua il suo task entro 100 ms (deadline).
- Molto probabilmente il sistema operativo non assegnerà la CPU immediatamente all'applicazione, poiché in un sistema multitasking ci sono altri programmi che competono per l'uso della CPU.
- Nella figura seguente è mostrato uno scenario in cui l'esecuzione dell'applicazione è stata completata entro la scadenza stabilita.

- istante di richiesta **R** = 0 ms;
- tempo di esecuzione **E** = $E1 + E2 = 20 + 40 = 60$ ms
- tempo I/O = 20 ms
- deadline **D** = 100
- tempo di attesa = $A1 + A2 = 10 + 5 = 15$ ms
- tempo di risposta = 90 ms



- Nei SO real-time generalmente i task hanno diverse criticità e quindi hanno diverse priorità.
- Le priorità possono essere assegnate in modo statico (**priorità statiche**) o calcolate dinamicamente in base alle caratteristiche dei singoli task (**priorità dinamiche**).
- I SO real-time sono classificati in ***hard real-time e soft real-time***. In questi ultimi, a differenza dei primi, una deadline non rispettata, non danneggia il funzionamento dell'ambiente operativo, ma ne abbassa le prestazioni e quindi la ***qualità del servizio*** (QoS *Quality of Service*).

Sistemi embedded Real-Time

- I computer embedded sono la forma più diffusa di computer.
- Questi dispositivi, spesso chiamati ***micro computer*** o ***micro controller***, sono usati in molti ambienti che comprendono automobili, aeroplani, robot industriali, sistemi di controllo per la domotica etc.

- Con questi sistemi, ad esempio, le case possono essere informatizzate e controllate, in modo da gestirne il riscaldamento, l'illuminazione, i sistemi di allarme e altro.
- Essi hanno compiti molto specifici e quindi spesso i sistemi operativi forniscono funzionalità limitate.
- Di solito, l'interfaccia utente è molto semplice o a riga di comando.
- In alcuni di essi sono installati sistemi operativi d'uso generale come Linux o Windows, in cui alcune parti del sistema, come ad esempio lo scheduler, sono modificate opportunamente per far girare applicazioni real-time.
- L'uso di sistemi embedded continua ad espandersi, sia come unità *stand alone* che come nodi connessi alla rete Internet e gestiti via web.
- L'accesso al Web può consentire a un utente di monitorare e controllare *da remoto* i dispositivi e lo stato dell'abitazione.

Sistemi operativi per personal computer

- I primi SO per personal computer erano semplici mono programmati e monoutente.
- Con l'aumento delle prestazioni dei microprocessori e delle dimensioni della memoria, la tecnica della multiprogrammazione è stata implementata anche nei SO per PC.
- Tutti i sistemi operativi moderni per PC sono multitasking: windows XP/7/8/10 (Microsoft), MacOS X di Apple e Linux che è anche un sistema time-sharing.

Operazioni del sistema operativo

- I sistemi multiprogrammati richiedono che diversi programmi siano allocati contemporaneamente in memoria.
- Per avere diversi programmi in memoria allo stesso tempo è necessaria una qualche forma di **gestione della memoria**.
- Inoltre, se alcuni programmi sono pronti per essere eseguiti allo stesso tempo, il sistema deve scegliere quale programma sarà eseguito per primo. Questa decisione è presa da una funzione del kernel, lo **scheduling della CPU**.
- In un sistema time-sharing, il sistema operativo deve garantire un tempo di risposta brevi. Il metodo più comune per garantire tempi di risposta brevi si ottiene con la **memoria virtuale**, una tecnica che permette l'esecuzione di un processo anche se non è completamente caricato in memoria. Il vantaggio principale della tecnica della memoria virtuale è che consente di eseguire programmi che sono più grandi della memoria fisica effettiva.
- Un sistema time-sharing deve anche fornire un file system. Il file system risiede su uno o più dischi e/o altri supporti di memorizzazione, di conseguenza, è necessaria la **gestione della memoria secondaria**.

- Inoltre, un sistema time-sharing fornisce un meccanismo per proteggere le risorse da uso inappropriato.
- Per garantire un'esecuzione ordinata, il sistema deve fornire i meccanismi per **la sincronizzazione e la comunicazione** dei processi, e assicurare che questi non restino per sempre bloccati in una situazione di **stallo**.
- Poiché il sistema operativo e i programmi utente condividono le risorse hardware e software del calcolatore, bisogna fare in modo che un errore in un programma utente non causi problemi ad altri programmi. Un programma erroneamente potrebbe modificare il codice o i dati di un altro programma, o anche il codice del sistema operativo stesso. Un sistema operativo progettato correttamente deve garantire che un programma non corretto, o volutamente dannoso, non possa modificare l'esecuzione di altri programmi.

Funzionamento dual-mode e multimode

- Per garantire la corretta esecuzione del sistema operativo, è necessario distinguere tra l'esecuzione di codice del sistema operativo e codice dei programmi utente.
- Per risolvere questo problema si utilizzano processori con un supporto hardware che permetta di differenziare tra i vari modi di esecuzione.
- Sono necessarie almeno due modalità di funzionamento: la **modalità kernel** (detta anche modalità privilegiata, di sistema o supervisore) e **modalità utente**.
- Un bit, detto **bit di modalità**, è presente nella CPU per stabilire la modalità di funzionamento corrente: kernel (0) o utente (1).
- Quando la CPU esegue un processo utente, il sistema è in modalità utente. Tuttavia, quando un'applicazione utente richiede un servizio dal sistema operativo, tramite una chiamata di sistema, il sistema deve passare da utente a modalità kernel.

- Questa tecnica di protezione si ottiene suddividendo le istruzioni macchina in istruzioni privilegiate e istruzioni non privilegiate.
- L'hardware della CPU consente di eseguire le istruzioni privilegiate solo in modalità kernel. Se si tenta di eseguire un'istruzione privilegiata in modalità utente, la CPU non esegue l'istruzione, ma piuttosto la considera come illegale e genera un'eccezione che sarà gestita dal sistema operativo.
- Ogni CPU ha il suo set di istruzioni privilegiate. Ad esempio, l'istruzione per passare alla modalità kernel è un'istruzione privilegiata. Alcuni altri esempi includono le istruzioni per il controllo dell'I/O, per la gestione del timer e dell'interrupt.
- Il concetto di modalità può essere esteso oltre due modalità (nel qual caso la CPU utilizza più di un bit per impostare e testare la modalità).
- La mancanza di una modalità hardware dual-mode può causare gravi problemi in un sistema operativo.

- Un programma utente che esegue istruzioni errate può mandare in crash il sistema operativo, scrivendo, ad esempio, in locazioni di memoria appartenenti al codice o ai dati del sistema operativo.
- I sistemi operativi come Microsoft Windows , Unix e Linux sono scritti per processori che utilizzano la funzionalità dual-mode e anche multi-mode.

Timer

- I progettisti, devono assicurarsi che il sistema operativo mantenga il controllo sulla CPU. Non si deve permettere che un programma utente resti in esecuzione in un ciclo infinito o non restituisca il controllo al sistema operativo.
- Per risolvere questo problema, si utilizza un timer. Un timer può essere impostato per generare un'interruzione dopo un determinato periodo. Prima di passare il controllo al programma utente, il sistema operativo imposta il timer affinché generi un'interruzione allo scadere di un certo periodo. Quando il timer genera l'interruzione, il controllo della CPU ritorna automaticamente al sistema operativo.

- Chiaramente, le istruzioni che modificano il valore del timer sono privilegiate.

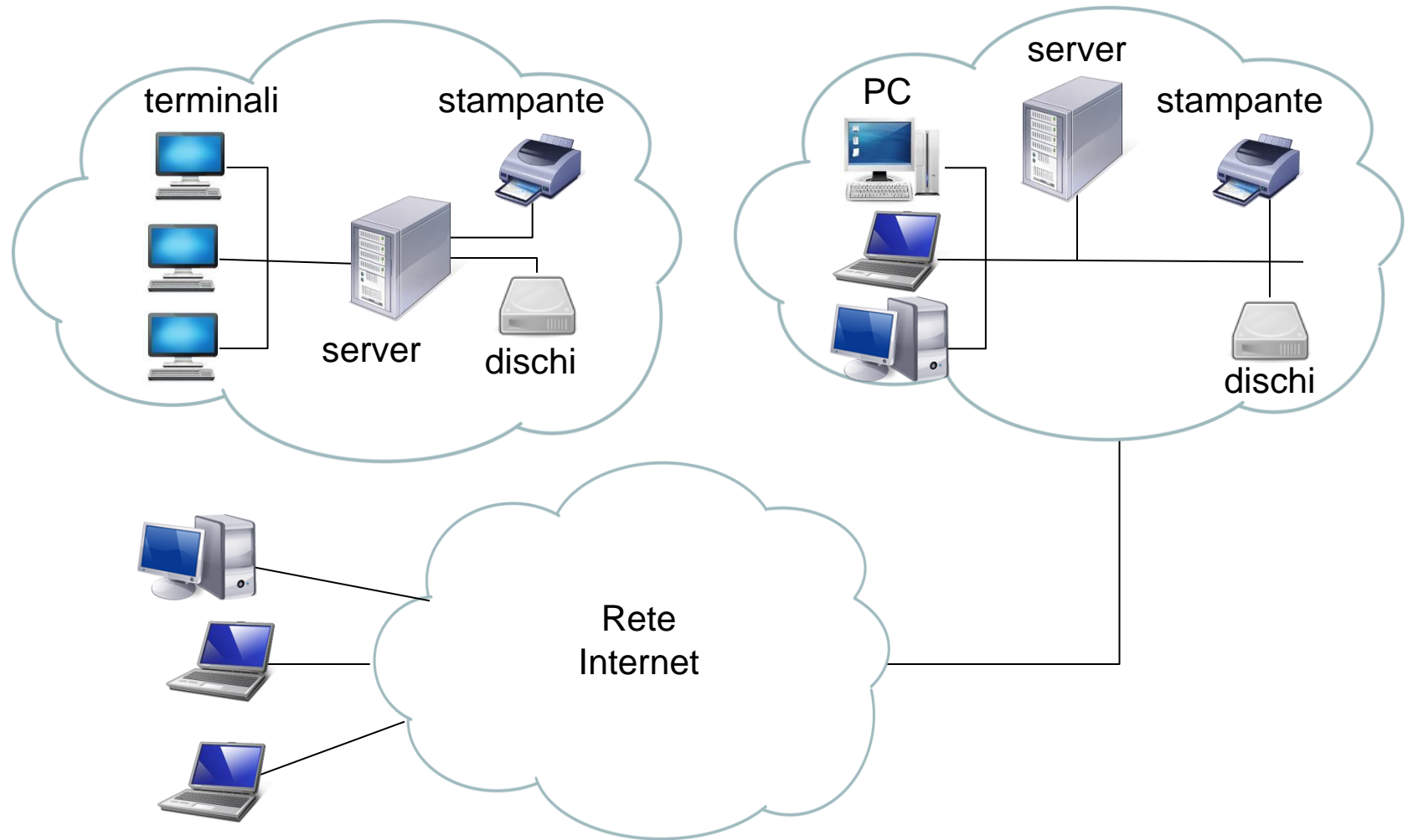
Ambienti di elaborazione

- Finora, abbiamo brevemente descritto alcune tipologie di sistemi operativi. Passiamo ora a spiegare come i sistemi operativi sono utilizzati in vari ambienti di elaborazione.

Computing tradizionale

- Solo pochi anni fa, un ambiente di elaborazione consisteva in un certo numero di PC collegati in rete, con un server che forniva servizi di condivisione di file e di stampa. L'accesso remoto era lento, limitato alla connessione tramite linea telefonica commutata con modem a 56 Kbit/s. Terminali collegati ai mainframe erano spesso usati in molte società e università.
- Con l'avvento della rete Internet, la tendenza attuale è di fornire varie modalità di accesso a questi ambienti informatici.
- Grazie alle tecnologie web e all'incremento della larghezza di banda WAN, le aziende forniscono portali web per accedere ai propri server interni.
- Computer desktop o portatili hanno sostituito i vecchi terminali.

- I computer mobili possono anche connettersi a reti wireless e alle reti dati cellulari per utilizzare il portale web della società.



Computing Mobile

- Il mobile computing si riferisce all'elaborazione su tablet e smartphone. Negli ultimi anni, tuttavia, la potenza di calcolo dei dispositivi mobili è talmente aumentata, in termini di funzionalità e prestazioni, tanto da rendere difficile la distinzione tra computer portatili e tablet.
- Oggi, i sistemi mobili sono utilizzati non solo per la posta elettronica e la navigazione su web ma anche per molte altre funzioni come la riproduzione di musica e video e la lettura di libri digitali.
- Molti sviluppatori ora realizzano applicazioni basate sulle caratteristiche integrate dei dispositivi mobili, come il sistema di posizionamento globale (GPS), accelerometri, giroscopi e altri sensori.
- Per fornire l'accesso ai servizi on-line, i dispositivi mobili in genere utilizzano sia le reti wireless wi-fi (IEEE 802.11) che le reti dati dei cellulari. La capacità di memoria e la velocità di elaborazione di dispositivi mobili, tuttavia, sono più limitati rispetto a quelle dei desktop.

I sistemi operativi che attualmente dominano il mobile computing sono Google Android, Apple iOS e Windows phone 10.

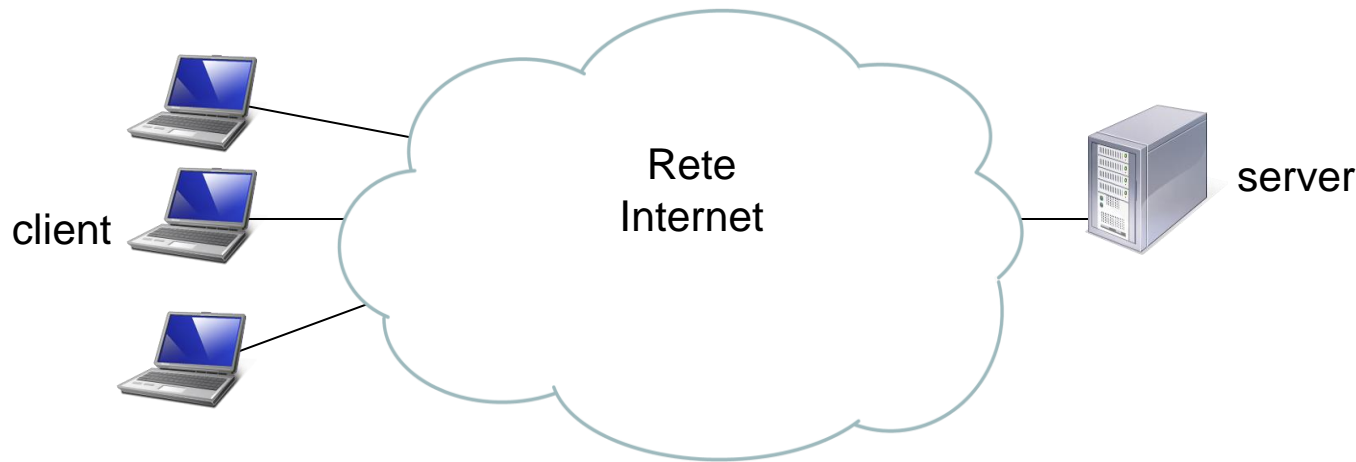
Sistemi paralleli e distribuiti

- I **sistemi paralleli** sono dotati di più CPU. In questi sistemi più processi potrebbero essere realmente eseguiti contemporaneamente.
- Il SO deve risolvere vari problemi legati all'esecuzione in parallelo dei programmi, nel caso che essi accedono simultaneamente agli stessi dati.
- Deve prevenire possibili condizioni di inconsistenza sulle strutture dati del sistema, che si potrebbero verificare se queste fossero modificate contemporaneamente da funzioni di SO eseguite su diverse CPU.
- Lo sviluppo delle reti ha portato alla realizzazione di moduli di SO per la gestione delle schede di rete e di vari servizi che hanno permesso ai sistemi collegati in rete di condividere dati e informazioni in vario modo come ad esempio tramite il web, e alla possibilità di realizzare applicazioni distribuite.

- I SO per calcolatori collegati in rete sono generalmente classificati in due categorie:
 - ***Sistemi operativi di rete***: ogni nodo della rete dispone di un proprio sistema operativo (non necessariamente uguale). I sistemi cooperano tra di loro, per esempio in applicazioni web a n-strati.
 - ***Sistemi operativi distribuiti***: in ogni nodo è installato lo stesso SO. L'utente vede l'intero sistema come un unico calcolatore. Il SO deve provvedere a bilanciare il carico di lavoro distribuendolo opportunamente sui vari nodi del sistema. I nodi condividono le risorse. Un esempio è dato dai sistemi **cluster**.

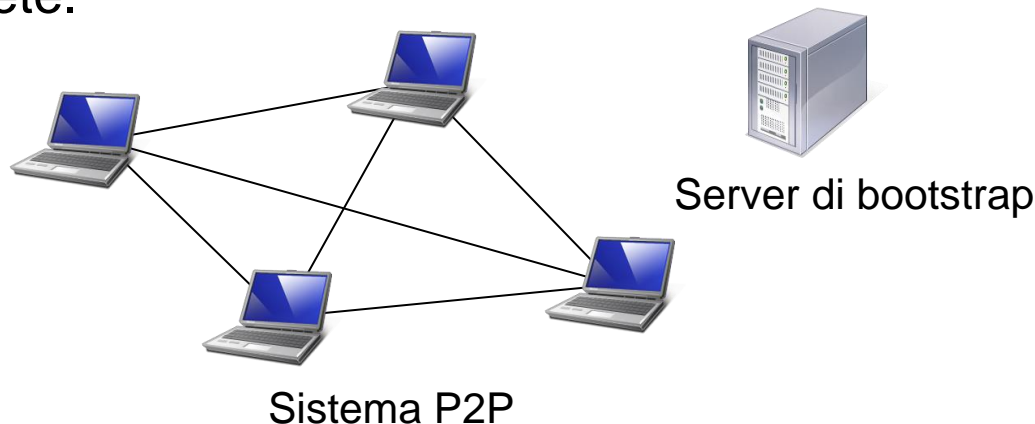
Client-Server

- Da quando i PC sono diventati più veloci, più potenti e meno costosi, i progettisti hanno abbandonato l'architettura di sistema centralizzato. Terminali collegati a sistemi centralizzati sono ora sostituiti da PC e dispositivi mobili dotati d'interfaccia grafica che si connettono ai server prevalentemente attraverso un'applicazione web.
- Di conseguenza, molti dei sistemi attuali funzionano come sistemi server per soddisfare le richieste generate da sistemi client. Questa forma di sistema, detto sistema **client/server**, ha la struttura illustrata nella figura seguente.



Computing Peer- to-Peer

- Un altro modello di sistema è il **peer- to-peer (P2P)**. In questo schema, tutti i nodi all'interno del sistema sono considerati pari, ciascuno può funzionare sia da client che da server. Ogni nodo può sia offrire che richiedere servizi.
- I sistemi Peer-to-peer offrono un vantaggio rispetto ai sistemi client/server. In un sistema client/server, il server è l'unico nodo a fornire servizi, mentre in un sistema peer-to-peer, i servizi possono essere forniti da diversi nodi distribuiti in tutta la rete. Per partecipare a un sistema peer-to-peer, un nodo deve prima associarsi alla rete di pari. Una volta che un nodo è connesso alla rete, può iniziare a fornire servizi e richiedere servizi da altri nodi nella rete.



Cloud Computing

- Il cloud computing è un'architettura d'elaborazione che offre servizi di rete come memorizzazione di dati e applicazioni. Ad esempio, alcuni fornitori di Cloud hanno migliaia di server, milioni di macchine virtuali e immense quantità di memoria secondaria disponibili per gli utenti di Internet. Gli utenti pagano tariffe in base ai servizi e alle risorse che utilizzano. Ci sono molti tipi del cloud computing, tra cui:
 - **Cloud privato** gestito da una società per uso proprio.
 - **Cloud pubblico**, un cloud disponibile via Internet per gli utenti che sottoscrivono un abbonamento per i servizi.
 - **Cloud ibrido** che include componenti di cloud pubblici e privati.
 - **Infrastructure as a Service (IaaS)**, server o dispositivi di memorizzazione utilizzabili, ad esempio, per fare il backup dei dati di produzione, fruibili attraverso Internet.
 - **Platform as a Service (PaaS)**, un ambiente software per sviluppare applicazioni e utilizzabile tramite Internet, ad esempio, web e database server.

- ***Software as Service (SaaS)***, una o più applicazioni, come word processor e fogli di calcolo, disponibili via Internet.
- In molti tipi di infrastrutture cloud, sono presenti sistemi operativi noti come Unix, Linux e Windows.